안전한 재구성 로봇 시스템: 설계, 프로그래밍 및 반응형 경로 계획

Safe Reconfigurable Robot Systems: Design, Programming, and Reactive Motion Planning

Alchan Yun

Submitted in Partial Fulfillment of The
Requirements for The Degree of

Doctor of Philosophy

in the
Department of Mechanical and Aerospace Engineering

at
Seoul National University

August 2020

ABSTRACT

Safe Reconfigurable Robot Systems: Design, Programming, and Reactive Motion Planning

by

Alchan Yun

Department of Mechanical and Aerospace Engineering
Seoul National University

The next generation of robots are being asked to work in close proximity to humans. At the same time, the robot should have the ability to change its topology to flexibly cope with various tasks. To satisfy these two requirements, we propose a novel modular reconfigurable robot and accompanying software architecture, together with real-time motion planning algorithms to allow for safe operation in unstructured dynamic environments with humans.

Two of the key innovations behind our modular manipulator design are a genderless connector and multi-dof modules. By making the modules connectable regardless of the input/output directions, a genderless connector increases the number of possible connections. The developed genderless connector can transmit as much load as necessary to an industrial robot. In designing two-dof modules, an offset between two joints is imposed to improve the overall integration and the safety of the modules.

To cope with the complexity in modeling due to the genderless connector and multi-dof modules, a programming architecture for modular robots is proposed. The key feature of the proposed architecture is that it efficiently represents connections of multi-dof modules only with connections between modules, while existing architectures should explicitly represent all connections between links and joints. The data structure of the proposed architecture contains properties of tree-structured multi-dof modules with intra-module relations. Using the data structure and connection relations between modules, kinematic/dynamic parameters of connected modules can be obtained through forward recursion.

For safe operation of modular robots, real-time robust collision avoidance algorithms for kinematic singularities are proposed. The main idea behind the algorithms is generating control inputs that increase the directional manipulability of the robot to the object direction by reducing directional safety measures. While existing directional safety measures show undesirable behaviors in the vicinity of the kinematic singularities, the proposed geometric safety measure generates stable control inputs in the entire joint space. By adding the preparatory input from the geometric safety measure to the repulsive input, a hierarchical collision avoidance algorithm that is robust to kinematic singularity is implemented.

To mathematically guarantee the safety of the robot, another collision avoidance algorithm using the invariance control framework with velocity-dependent safety constraints is proposed. When the object approached the robot from a singular direction, the safety constraints are not satisfied in the initial state of the robot and the safety cannot be guaranteed using the invariance control. By proposing a control algorithm that quickly decreases the preparatory constraints below thresholds, the robot re-enters the constraint set and avoids collisions using the invariance control framework.

The modularity and safety of the developed reconfigurable robot is validated using a set of simulations and hardware experiments. The kinematic/dynamic model of the assembled robot is obtained in real-time and used to accurately control the robot. Due to the safe design of modules with offsets and the high-level safety functions with collision avoidance algorithms, the developed reconfigurable robot has a broader safe workspace and wider ranger of safe operation speed than those of cooperative robots.

Keywords: Reconfigurable robots, Modular robot software, Directional safety, Reflected mass, Collision avoidance, Singularity avoidance.

Student Number: 2010-23223

Contents

List of Tables

List of Figures

1 Introduction

Unlike robots currently employed in today's factories, which are of a fixed design and must operate in areas in which human access is restricted, the next generation of robots are increasingly being asked to work in close proximity to humans, even interacting with humans and other robots in collaborative tasks. The trend in manufacturing toward smaller batches of customized products also requires different types of robots, e.g., SCARA architectures are well-suited for tabletop assembly tasks, while six- and seven-dof elbow-type (PUMA) robots are better suited for more complex three-dimensional assembly and other tasks.

1.1 Modularity and Reconfigurability

Modular reconfigurable robots, in which various link and actuator modules can be manually connected into robots of different topologies, offer one intriguing solution to the flexibility requirement. Rather than purchasing and installing different types of expensive fixed architecture robots (whose installation and reconfiguration are

also tedious and time-consuming), modular reconfigurable robots can, in principle, be quickly reconfigured into different topologies that are suited for the task at hand.

Given these advantages, modular reconfigurable robots have a long history and extensive literature. Since the development of the Dynamically Reconfigurable Robotic System (DRRS) [1] and Reconfigurable Modular Manipulator System (RMMS) [2] in 1988, various types of modular robots have been developed. According to the recent survey on modular reconfigurable robots [3], manipulator type modular robots can be broadly classified into two major categories: self-reconfigurable robots and manually reconfigurable manipulators. Because the self-reconfigurable robots do not require manual reassembly by humans, the reconfiguration is performed easily and quickly. However, these robots have separate motors and related parts for self-reconfiguration, which take up more space and mass, thus reducing the overall practicality of the robot. For industrial applications, manually reconfigurable manipulators are more practical than self-reconfigurable manipulators.

Attempts have been made to develop manually reconfigurable modular manipulators that are well suited for industrial applications. Fig 1.1 shows pictures of manually reconfigurable modular robots that have been developed so far. The Reconfigurable Modular Manipulator System (RMMS) [2, 4, 5] is one of the earliest implemented manipulators of this kind. The main feature of RMMS is that the connectors for the modules can transmit electrical power and communication signals without custom cabling. One type of pivot joint module is developed, and a three-dof robot is constructed by connecting three joint modules. Schunk commercialized the Lightweight Arm (LWA) modular manipulator [6], which consists of two-dof joint modules called *Powerballs*. Since one module is designed to have two degrees of freedom, a six-dof manipulator with a high weight-torque ratio can

(a) (b) (c) (d)

Figure 1.1: Existing modular manipulators. (a) RMMS, (b) LWA-4P, (c) RMM, (d) MARA.

be constructed by connecting three modules. However, it is relatively difficult to reassemble because the coupling is performed by connecting several bolts. In addition, the parts for the mechanical coupling are bulky. The Reconfigurable Modular Manipulator (RMM) [7] consists of two types of one-dof joint modules: rotate and pivot. By combining the joint modules, several types of manipulators are configured including a seven-dof elbow-type robot. Acutronic Robotics recently launched the Modular Articulated Robotic Arm(MARA) with Hardware Robot Operating System (H-ROS) [8]. The distinguishing feature of MARA is that each module can be directly integrated with Robot Operating System (ROS) and a simplified communication interface with other ROS components is available. In terms of hardware, MARA has features similar to Powerball: two-dof joint modules and bulky mechanical coupling.

As described above, the common disadvantage of existing modular reconfigurable robots is that they are often bulky, with limited payload and force/torque

generation capabilities because of limited power transfer between modules. In addition, since their modules can be assembled only in a predetermined order and direction, they do not adequately reflect the advantages of modular robots that can be reassembled in various forms. The few commercial modular reconfigurable robots that have been developed, LWA-4P and MARA, have fallen far short of the needed performance requirements for practical use.

One method to increase the modularity is to adopt genderless connectors. By being able to connect the modules regardless of the input/output directions, the number of possible connections can be greatly increased. In industrial applications, the important aspects of the genderless connector design are the stability and the capacity of the electrical and mechanical connection. In a recent survey on coupling mechanisms for modular reconfigurable robots [9], several genderless coupling mechanisms [10–16] support both mechanical and electrical connections. The electrical connections are achieved by placing spring pins to have rotational symmetry about the coupling axis. However, due to the small number of electrical contacts, the connectors do not transmit enough electricity to drive industrial robots. The mechanical connections are also not designed to withstand the forces exerted on the joint axis of industrial robots.

1.2 Safe Interaction

Today's industrial robots are applied mostly as part of fixed automation lines, e.g., machines that perform pre-programmed motions for tasks such as assembly, welding, painting, and other tasks that require only small corrective measures against disturbances, measurement errors, and other sources of deviation from a user-specified desired trajectory. In contrast, the next generation of robots are being designed to

perform collaborative tasks, with humans as well as other robots, in less-structured, complex environments. Such task requirements require first and foremost that the robot operates in a safe manner, without causing injury to humans or other robots and objects in its environment. In [17] the basic requirements of safe robot interaction are outlined in detail; the most basic requirement is the ability to detect collisions, followed by the ability to anticipate collisions and to take preparatory action, e.g., to avoid the collision in a safe manner, or failing that, to minimize any potential damage caused by the impact of the collision. We refer the reader to the references cited in [17] for a survey of collision detection methods.

The literature on collision avoidance is also extensive. For real-time reactive planning in dynamically changing environments, methods based on the artificial potential field of Khatib [18] have been by far the most successful and popular. In [19], the potential field method is extended to redundant robots so that a robot can avoid collisions while executing the original task. In [20], a torque-based self-collision avoidance algorithm is proposed with careful design of damping term in the repulsive forces. A depth space approach is introduced in [21], which utilizes low-cost depth sensors for collision avoidance. In [22], estimation of the obstacle's velocity is taken into consideration when formulating the repulsive potential function. In [23], an invariance control framework is utilized for collision avoidance by considering the potential function as a constraint output function. While the above potential field planning methods often require careful design and tuning of the potential function to avoid, e.g., local minima, kinematic singularities, and zig-zag paths, by appropriately designing and combining an attractive and repulsive potential field, a robot can be smoothly guided to the desired goal position while avoiding obstacles appearing in its path.

The literature on anticipatory and preparatory actions for minimizing damage

from potential collisions is more recent. A well-known measure of how much a robot is prepared for a particular direction is *the reflected mass* [24], which is a function of the robot's posture and direction of interest. Depending on the related ellipsoid and formulation, it is also called the force transmission ratio [25, 26], admissible acceleration radius [27], and inverse of reflected mass [28]. The reflected mass is widely used in the literature as a criterion for determining safety. In [24], the reflected mass is used to prove that serially connecting a large-sized robot to a small-sized robot is safer than using only a small-sized robot. In [28–30], to reduce the amount of impact in an unexpected collision, the robot is moved to a posture with a small reflected mass through the null motion of the redundant robots. In [31–33], the robot is able to move at a higher speed while satisfying the impact threshold by reducing the reflected mass in the moving direction. These studies have one thing in common: they control the robot in a direction that reduces the reflected mass of the robot in a specific direction.

Since collisions between links and objects occur in the task space, the prerequisite in the collision avoidance algorithm is that the robot must have the ability to move in the direction of interest. However, if the robot is in a kinematic singularity and the object approaches the robot in the singular vector direction as shown in Fig 1.2, the robot cannot generate the control input in the direction of reducing the risk the fastest.

To escape from this situation, one can think of moving the robot in a direction that reduces the reflected mass. However, there is an issue in using the reflected mass in real-time applications: the gradient of the reflected mass becomes numerically unstable when the direction of interest is close to the principal axes directions of the ellipsoid matrix. In the worst-case scenario illustrated in Fig 1.2, the value of the reflected mass becomes infinite. However, the gradient of the reflected mass

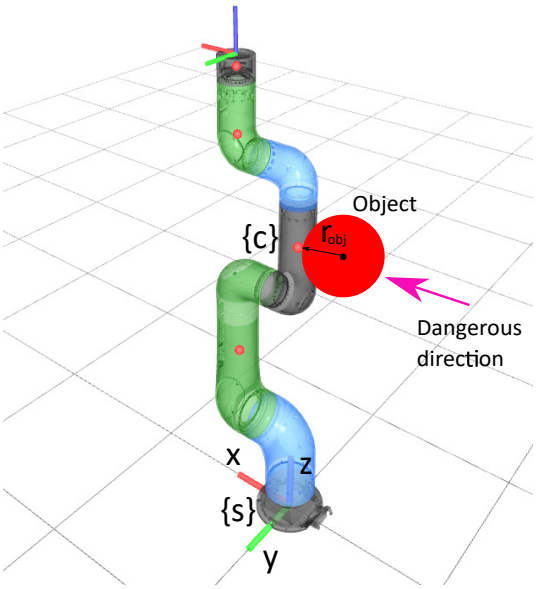


Figure 1.2: Example of a collision occurring at a kinematic singularity.

is also infinite, so it is not well-defined and needs numerical handling. Because of this numerical instability, the reciprocal of the reflected mass is often used, but this also has a gradient value of zero, making the robot impossible to escape from a singular configuration. A similar issue is addressed in [33] involving a redundant robot: this issue is bypassed by defining a new concave attractive potential function by finding the minimum value of the reflected mass in the entire null space. However, this method is only applicable to redundant robots and is numerically complex.

The most straightforward way for a robot to remain safe is to maintain a certain level of manipulability for all directions so it can react to any external danger. Some of the previous works along this line include a term increasing the general isotropic kinematic manipulability [34–36]. This remedy, however, can constrain

the workspace of the robot to a limited volume. Furthermore, the resultant posture of the robot might not be optimized for the current object state since the measures are functions of the robot's posture only.

One important fact is that the matrix defining the ellipsoid of the reflected mass is inversely related to the matrix defining the dynamic manipulability ellipsoid as described in [28]. In other words, having a small reflected mass in a direction means that the robot has a high manipulability in that direction. So the problem of reducing the reflected mass can be replaced with the problem of increasing the directional manipulability. It is possible to define a virtual ellipsoid having a first principal axis in the direction of interest and to use a method of controlling the robot such that the manipulability ellipsoid of the robot approaches to the virtual ellipsoid. From the fact that the matrix defining the manipulability ellipsoid is positive (semi-)definite, we can define the distance between two positive (semi-)definite matrices as a new safety measure. Since the space of positive (semi-)definite matrices is different from vector space, it is necessary to reflect the geometric properties in obtaining the distance. In [37], a similar method has been proposed for the case of the space of $n \times n$ positive-definite matrices, $P(n)$. A distance on $P(n)$ is used to represent the discrepancy between two ellipsoids and the robot is controlled to the desired manipulability ellipsoid by controlling the robot in a direction to reduce it. Although this method is geometrically complete, the distance on $P(n)$ becomes numerically sensitive when the robot is close to the kinematic singularity, because it uses the matrix inverse.

1.3 Contributions of This Thesis

In this thesis we address the two fundamental problems addressed above, with the goal of developing an efficient and practical reconfigurable modular robot system with basic safe interaction capabilities.

1.3.1 A Reconfigurable Modular Robot System with Bidirectional Modules

We first propose a novel reconfigurable modular robot system with improved performance capabilities with respect to force-torque generation and power efficiency, accuracy and repeatability, and improved modularity features. The main feature of our system is that industrial robots requiring high precision and high speed can be configured with modules equipped with genderless connectors. Unlike the connectors introduced earlier, the developed genderless connector can transmit as much power and signals as necessary to an industrial robot through a large number of electrical contacts. Mechanical coupling is designed to withstand the forces generated by the motion of the robot, while also having the genderless characteristic. With the genderless connector, several links and joint modules are developed. Since they can be connected regardless of input and output, we call them bidirectional modules.

Another feature of the proposed system is that it consists of joint modules with two-dof. Existing two-dof modules such as Powerball and MARA are bulky because the two joint axes are designed to meet at one point. By imposing an offset between the two axes, it is possible to connect the two axes with the minimum number of components, thus enabling a lightweight and compact design. This offset not only helps to improve the overall integration but also helps to improve the

safety of the module; it serves to increase the robot's singularity-free workspace. In addition, a flexible component is added to the joint, which is flexible in the direction of the rotation axis, so that it can absorb shocks occurring from collisions.

1.3.2 A Modular Robot Software Programming Architecture

In addition to the hardware design, another key factor to turning the reconfigurable modular robot into a practical tool is an easy-to-use, efficient programming architecture. The architecture for the modular robot consists of three parts: a data structure for each module, the connection representation, and the kinematic/dynamic modeling of resultant configuration. In [38], the kinematics of the modular robot is expressed by the Denavit-Hartenberg(D-H) parameter expression. However, this method is not efficient because the link frames must be newly defined every time the connection is changed. In [39], kinematics and dynamics for modular robots are expressed in modern screw theory. They represented the connections in terms of Assembly Incidence Matrix (AIM) and the hierarchy in terms of Accessibility Matrix (AM). In [40], a method of synthesizing the inertia components assigned to each joint has been described. A common feature of the aforementioned connection representations is that they always represent the connection status between links and joints. If a joint module itself is a tree-structure with multiple joints, the representations includes the components that are not used in the resultant kinematic tree, making it inefficient. In [41], a data structure called adaptive robotic system architecture(ARSA) is introduced, which deals with more general types of modules and their connections. However, it does not consider which link component the connection port is assigned to, and modeling using the data structure is not performed.

We propose an efficient programming architecture to efficiently model the kinematics and dynamics of tree-structured modules with multi-dof and bidirectional connection ports. The main feature of this architecture is the hierarchical representation of connections, which consists of intra-module and inter-module relations. The intra-module relation describes the relationship between links, joints, and connection ports, which is not dependent on the connection. Once the connection between two modules is made, we define the inter-module relation to describe which kinematic branch is relevant to the current connection. Since the direction of the generated tree is also determined at this stage, it is possible to obtain the correct kinematic/dynamic model even when the connection direction is reversed by the bidirectional port.

1.3.3 Anticipatory Collision Avoidance Planning

The second major contribution of this thesis is a collection of methods for planning motions that anticipate and minimize the potential damage ensuing from any collisions. We consider two methods: in the first method, we assume the repulsive potential is dependent only on the relative position between an object and a robot, while in the second method we assume the repulsive potential is dependent on the relative velocity.

For the preparatory actions, we define novel direction-dependent safety measures with the following property: the measure should have non-zero and bounded norm of gradient with respect to joint angles for the entire configuration space and for all directions except for the local maximum.

The first safety measure is a modified reflected mass, which has a non-zero gradient at the local minimum of the directional manipulability. The measure is simply defined by a family of function compositions. The method is simple but

allows the robot to quickly escape from the local minimum. One advantage of the new measure is that it exhibits a physically intuitive safety since it shows a tendency similar to that of the reflected mass. However, this method still has an issue that the modified reflected mass has very small gradient when the direction of interest is close to the second principal axis direction.

The second safety measure is motivated by the positive (semi-)definiteness of the ellipsoid matrix. Increasing manipulability along a direction can be geometrically interpreted as controlling a robot to have a manipulability close to a target manipulability which has its major axis along the direction. Instead of using the distance on $P(n)$ as in [37], the distance on the space of positive semi-definite matrices with rank-1, $S^+(1,n)$, is proposed as a new safety measure. By setting the target matrix with only the components of the direction of interest and approximating the current robot's manipulator matrix to have only the main axis components, the distance on $S^+(1,n)$ can be used. The measure is a special case of more general closeness measure defined on $S^+(p,n)$, introduced in [42].

By generating control inputs in a direction that reduces the second safety measure, the robot approaches a safer posture. Unlike the case of the distance on $P(n)$, it is well-defined even when the robot is at a kinematic singularity. In addition, it also satisfies the requirement of a safety measure: it has non-zero and bounded gradients. It is also applicable even to systems with very low degrees of freedom since it requires only the major axis of the manipulability matrix. The second safety measure also has several advantages over directional manipulability: (1) it generally shows faster convergence to the solution posture, (2) the distance traveled until convergence is shorter, and (3) it is able to track the non-maximal manipulability while directional manipulability measures only aim the maximal target manipulability.

Using the proposed safety measures, we develop a kinematic collision avoidance algorithm that is robust to kinematic singularities. The algorithm combines a repulsive action, generated by conventional Cartesian space potential functions, and a preparatory action, to increase the directional manipulability along the direction in which the repulsive action is applied. A hierarchical control framework is used to properly blend the avoidance motion and the operation of performing the nominal task. The essence of this algorithm is that it adds another layer that blends repulsive motion with preparatory motions.

For velocity-dependent repulsive potentials, the invariance control framework [43] is used to avoid collision and, at the same time, mathematically guaranteeing safety. To intuitively apply the invariance control to collision avoidance, the velocity-dependent potential functions of [44,45] are used for repulsive constraints. Since the invariance control framework does not guarantee the safety of the robot when the initial state of the robot is not in the invariant set, a control method is proposed to escape from such cases using output linearization of the preparatory constraint and simple optimization using linear programming.

To generate a stable control input around the kinematic singularity, the gradient of the proposed safety measures must have an analytic expression and be able to be calculated in real-time. In this thesis, the requirements are achieved by using the Lie theoretic formulation of the kinematics/dynamics of robots introduced in [46].

1.4 Organization of This Thesis

In Chapter 2, we describe the hardware development of the Modular Manipulator (ModMan) system. We compare the performance of the proposed genderless connector to existing connectors to show that the proposed connector has sufficient electrical capacity to be used in a robot for industrial applications. Joint and link modules equipped with the genderless connector are introduced, and various types of robots that can be configured using these modules are also introduced. To prove that the assembled robot can perform many tasks performed by existing robots, we provide a result of a repeatability test for a six-dof configuration.

In Chapter 3, we describe a programming architecture for modular reconfigurable robot systems, including a data structure, a matrix representation of the connection of modules, and an automatic kinematic/dynamic modeling algorithm. The automatic kinematic modeling algorithm is validated with ModMan's hardware. To show the capability of the ModMan, a set of experiments including task space motion planning, admittance control using a force-torque sensor, and simple pick-and-place tasks with various vision sensors, are conducted.

In Chapter 4, we propose a collision avoidance algorithm with velocity control. Tow safety measures are proposed to control the robot to the position with higher manipulability in a desired direction. The control behavior of the proposed safety measure is validated using 2-D and 3-D simulation and compared with other safety measures and distance measures. The collision avoidance algorithm is validated using actual hardware, especially the case when the object approaches the robot from a singular direction.

Chapter 5 describe a collision avoidance algorithm with velocity-dependent constraints. We first formulate the invariance control with velocity-dependent potential functions. Then we propose a control method to extricate the robot from the state far from invariance states. Simulation results are provided to show that the proposed dynamic collision avoidance algorithm guarantees the safety of the robot.

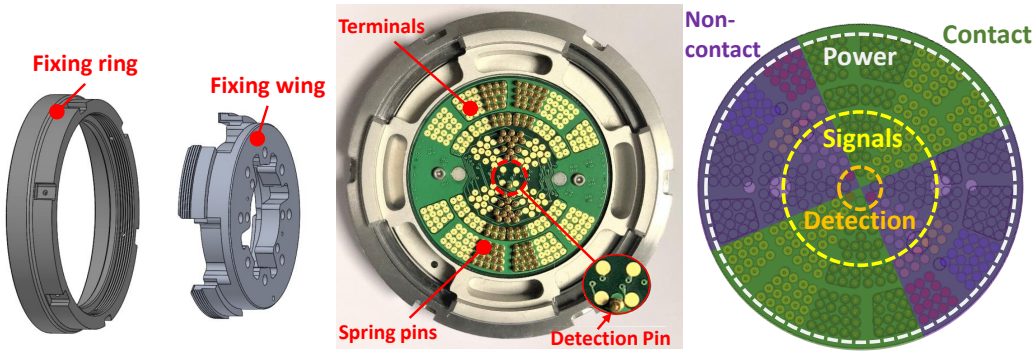
We conclude this thesis in Chapter 6 with a summary and an evaluation of the results.

2

Design and Prototyping of the ModMan

In this chapter, we introduce our newly developed modular manipulator system named *ModMan*. The main feature of the ModMan's hardware is the genderless connector, granted for US patent [47], which is designed to satisfy the mechanical/electrical requirements to be used in industrial robots. The developed modules equipped with the genderless connector are called *bidirectional modules*, which can be connected to other modules regardless of input/output directions. A set of two-dof joint modules and a one-dof joint module are built to configure six- and seven-dof elbow-type (PUMA) robots.

This chapter is organized as follows. Section 2.1 describes the developed genderless connector. Section 2.2 describes the developed joint and link modules. Section 2.3 shows experimental results regarding repeatability the ModMan. Section 2.4 concludes this chapter with further comments.



(a) (b) (c)

Figure 2.1: Genderless connector. (a) fixing ring and wing, (b) arrangement of spring pins and terminals, (c) electrical connection.

2.1 Genderless Connector

The genderless connector of ModMan consists of a mechanical connection mechanism and an electrical connection as illustrated in Fig. 2.1. The connector is designed to be assembled at four different angles, allowing the range of motion to be varied depending on the task at hand.

For an intuitive and rigid connection, a simple screw-type mechanical connector is employed as shown in Fig. 2.1a. The connector consists of a fixing wing with outer threads and a fixing ring with inner threads. As illustrated in Fig. 2.2, each fixing wing is fastened with the fixing ring of the other side. The fastened structure can support bending and torsional stresses in all directions. Note that the tolerance in the coupling between two fixing wings is sliding fit to minimize error and backlash. Although it is possible to tighten the fastener by hand, it is best to use a hook spanner to ensure a promising connection.

To ensure structural stability of the connector, stiffness of the fixing wing is

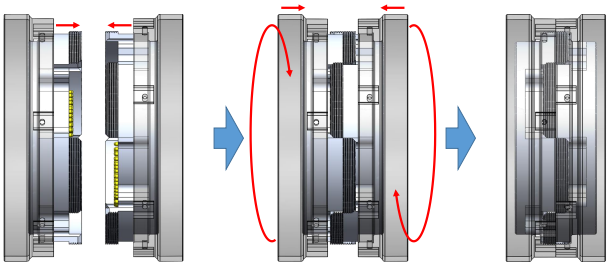
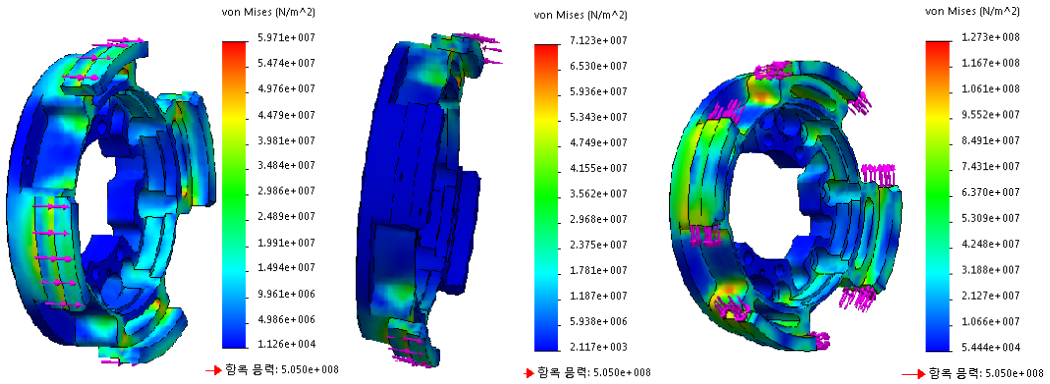


Figure 2.2: Connection diagram

analyzed by the finite element method(FEM) of SolidWorks, and the results are shown in Fig. 2.3. An extremely high tensile force of $1,000N$ is applied to the fixing wing and the resulting maximum stress is $5.97\times10^7N/m^2$ which has an associated factor of safety (FoS) of 8.46. A torsion of $160Nm$, the maximum peak torque of the large joint module (Table 2.2), is applied to the fixing wing and the resulting maximum stresses about the radial and axial directions are $7.12\times10^7N/m^2$ and $1.27\times10^8N/m^2$ which have associated FoSs of 7.09 and 4.97, respectively. Because the results are for a single fixing wing only, the actual FoS of the assembly of two fixing wings and rings are to be doubled at least. All the results ensure that the geometric structure of the mechanical connection is more than safe.

The electrical connection of the genderless connector (Fig. 2.1b) is responsible for transferring electrical power and communication signals between the modules. The contacts between spring pins and surface terminals are used for the electrical connection to give compliance for robustness against mechanical clearances. As illustrated in Fig. 2.1c, the electrical connection is divided into three sections: inner section for a detection pin to recognize offset angle, middle section for 26 signals including Ethernet for Control Automation Technology (EtherCAT) signals, and outer section for power and ground. To enable the connections at every 90° and



(a) (b) (c)

Figure 2.3: FEM result. (a) tensile force of 1000N, (b) torsion about radial direction of 160Nm, (c) torsion about axial direction of 160Nm.

to maximize the number of the contacts, the spring pins are arranged in two diametrically opposite octants. Note that the detection pin is located in only one of the octants for 90° resolution. Two sets of surface terminals are arranged in mirror symmetry of the spring pin arrangement, and at 45° intervals to the left and right of the pin arrangement. Based on this efficient placement, a total of 105 spring pins are mounted on the connector. The genderless connector can transmit more power and a greater number of signals than the connectors of other modular robots as shown in Table 2.1. The proposed genderless connector can support electrical loads of high-dof configurations such as a dual-arm configuration.

System	Type	Power	Signals	Number of Spring Pins
RMMS [4]	Gendered	72V-25A 48V-6A	2 RS-485, 4 Videos	30
Powerball [6]	Gendered	24V-15A	CANopen, 10 Signals	20
MARA [8]	Gendered	48V-8A	Ethernet	16
Thor [12]	Genderless	2 Pins	RS-485,	6
HiGen [15]	Genderless	4 Pins	I2C, Serial	12
ModMan (proposed)	Genderless	88 Pins (48V-56A 5V-48A)	EtherCAT, 22 Signals	115

Table 2.1: An electrical capacity comparison of connectors in reconfigurable modular robot systems.

2.2 Modules for ModMan

2.2.1 Joint Modules

To build the joint modules for ModMan, we develop three types of joints with similar structures and different performances. The design philosophy is that the joint has a high volume-power ratio. Figure 2.4 shows the structure of the joint with the largest power among the three joints. To have high power in a small space, we used a frameless motor with a hollow shaft. To have high accuracy and backdrivability, a harmonic drive is used as a reducer. By placing a torsion bar in the space inside the harmonic drive, the joint has additional joint flexibility for the

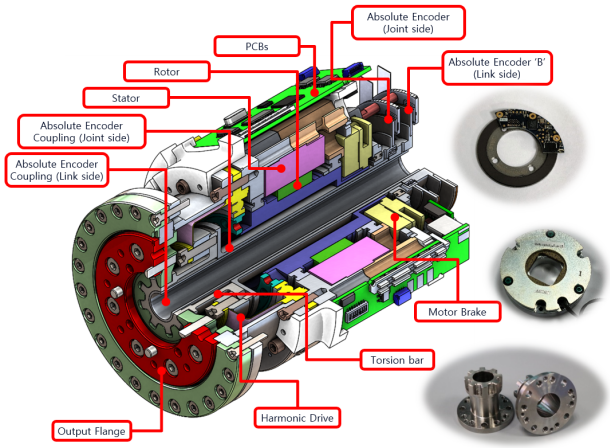


Figure 2.4: Details of two-dof module and joint module with regard to large size

safety of the robot. The added joint flexibility affects the dynamics of the robot, which will be covered in Chapter 3. Two encoders are used to measure deflection caused by the torsion bar. The genderless connector is attached to a flange at the output side of the motor. Detailed specifications of three types of joints are described in Table 2.2.

For effective management and development of the modules, three types of printed circuit boards(PCB) are developed in a modular manner; a controller PCB, an EtherCAT PCB, and a motor driver PCB. Since EtherCAT communication protocol has a directional property of data in/out ports, a switching circuit is implemented to change the direction of EtherCAT communication. The PCBs are designed to have the same geometry and connected by flex cables so that a set of PCBs can wrap around a joint in a manner of minimizing the required space for installing them.

To construct various types of manipulators, several types of joint modules with the genderless connector have been developed as shown in Fig 2.5. Two-dof joint

Model	**Large**	**Medium**	**Small**
Motor	TBMS-7615A	TBMS-6013A	RBE-01211A
Gear ratio	1 : 100	1 : 100	1 : 100
Rated torque	93.9 Nm	41.5 Nm	22.3 Nm
Peak Torque	157 Nm	82 Nm	54 Nm
Speed	42.26 rpm	54.2 rpm	103.47 rpm
Encoder	19 bit	19 bit	19 bit
Resolution	0.002 deg	0.002 deg	0.002 deg

Table 2.2: Specification of two-dof joint modules

modules are built by assembling two joints so that the two joint axes are orthogonal with offset. They are labeled as "typeA" and "typeB", depending on the position of the connection port. A total of six two-degree-of-freedom joint modules with three different sizes for each type are developed. Besides, a one-dof module is developed with the joint used for large size modules. It is commonly used to construct a seven-dof elbow-type manipulator by connecting it after the first two-dof joint module.

In two-dof joint modules, the offset between the two axes plays two roles: it reduces the overall volume of the module and increases the safe area within the workspace. Fig 2.6 describes how the joint offset affects the first two degrees of freedom of a robot. In many existing robotic manipulators, the axes of the first two joints intersect as shown in the figure on the left, while Modman's two-dof module has an offset as shown in the figure on the right. When there is no offset,

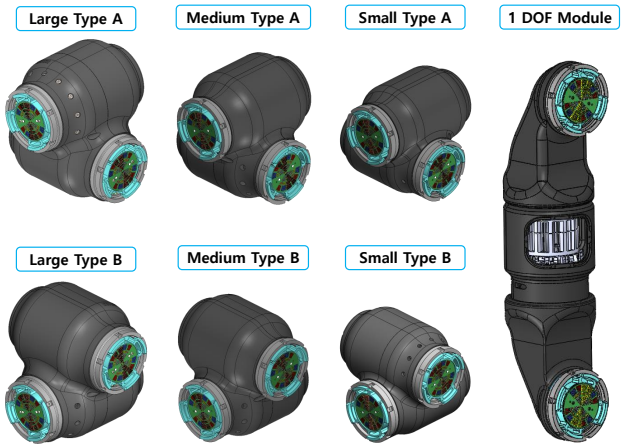


Figure 2.5: Two-dof joint modules and one-dof joint module

the translation part of Jacobian described in {c} is

$$J^c_{v,\text{left}} = \begin{pmatrix} L\cos q_2 & 0 \\ 0 & 0 \\ 0 & -L \end{pmatrix} \tag{2.2.1}$$

In contrast, if there is an offset a between the first and second axes as shown in the figure on the right, the Jacobian is

$$J^c_{v,\text{right}} = \begin{pmatrix} L\cos q_2 + a & 0 \\ 0 & 0 \\ 0 & -L \end{pmatrix}. \tag{2.2.2}$$

The yellow area represents the frontal workspace where the rank of J^c_v is 2. When a collision occurs, a high rank of J^c_v means that there are more directions in which the shock can be absorbed. When there is an offset, the yellow area is wider than when there is no offset, which means it is safer.

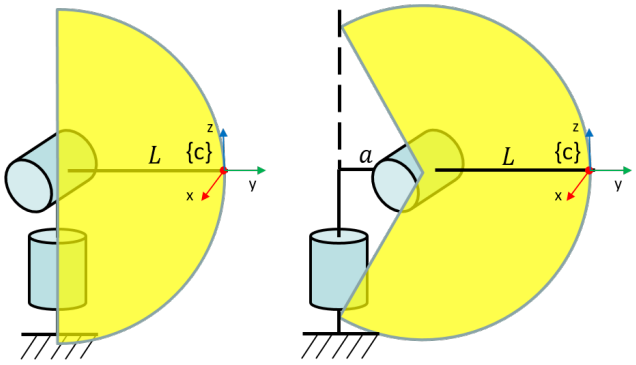


Figure 2.6: Effect of offset. Yellow area represents the frontal workspace with $Rank(J) = 2$. (left) without offset, (right) with offset.

2.2.2 Link and Gripper Modules

In addition to the joint modules, four types of link modules with the genderless connector have also been built as shown in Fig. 2.7. "Link module A" and "Link module B" are designed for upper and lower arms, respectively. Likewise, "Base link module" and "Torso" are designed for one-arm and human-like dual-arm configurations, respectively. Note that each link module also acts as a slave even though it does not have any actuators. This is to automatically obtain the module ID, connection direction, and offset angle. Thus, each link module has a controller PCB and an EtherCAT PCB. The weights of link modules A and B are $3.88Kg$ and $3.4Kg$, respectively.

For the proof of concept, we also designed a gripper module. The jaws of the gripper module are coupled to each other to be driven by a single actuator. To pick up soft objects, the gripper has a spring-based sensor that can measure the gripping force. The gripper also utilizes an RGB-D sensor (RealSense SR300) to perform vision-based pick-and-place tasks.

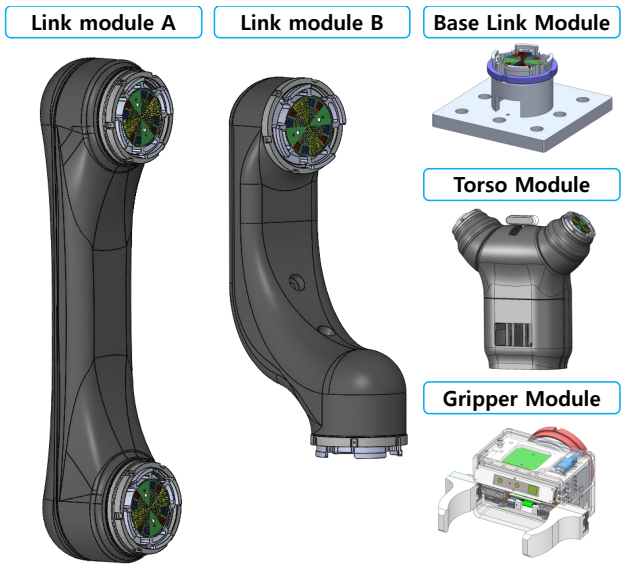


Figure 2.7: Link and gripper modules.

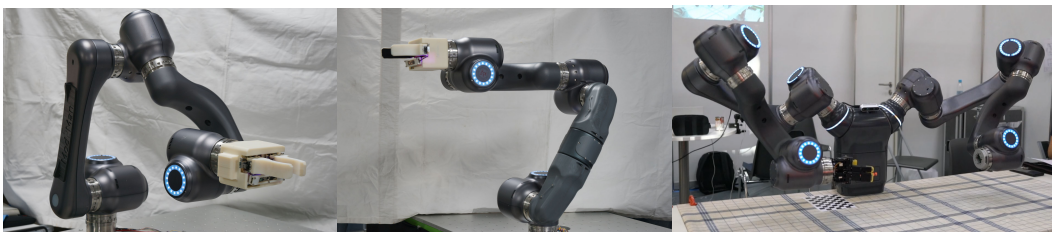
Examples of possible configurations with developed modules are shown in Fig. 2.8.

2.3 Experiments

We performed an accuracy evaluation test to prove that the repeatability of the ModMan, assembled in a six-dof configuration, is comparable to that of a non-reconfigurable robot.

2.3.1 System Setup

To validate the performance of the ModMan compare to conventional manipulators, we develop software for low-level controls and high-level commands. The slave controller performs computations related to the EtherCat communication, sensor



(a) (b) (c)

Figure 2.8: Examples of different configurations. (a) six-dof, (b) seven-dof, (c) dual arms.

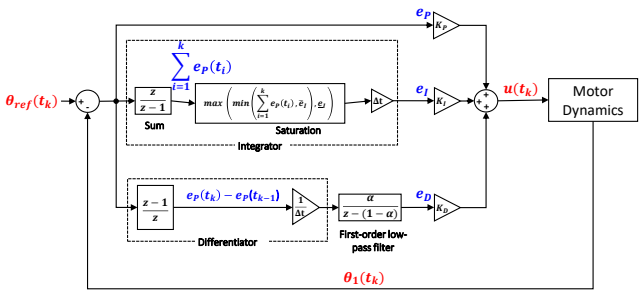
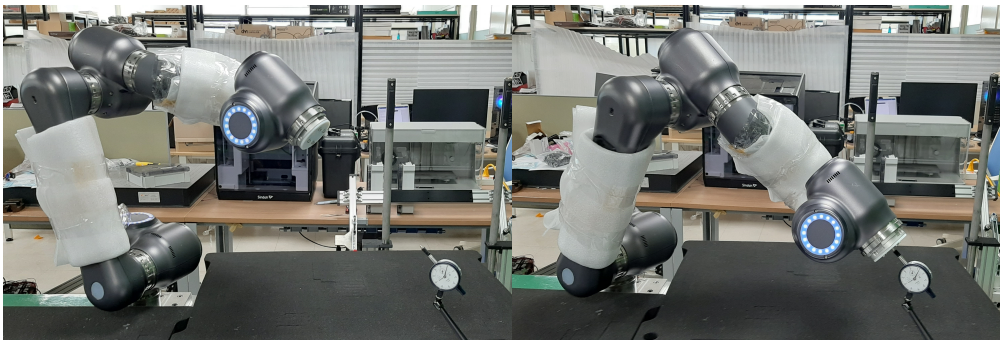


Figure 2.9: Illustration of the PID controller used in the joint modules

signals, and joint inputs. All the sensor data are collected through SPI multi-drop communication. For the EtherCAT communication between slave modules and the master controller, the communication protocol follows CoE (CANOpen Over EtherCAT) DS 402 stack which is widely used as a standard for CAN-based systems.

For the independent joint control, a discrete PID controller for position control is implemented as described in Fig. 2.9. The output is a link-side angle and the motor input is pulse width modulation (PWM) whose frequency is set to be 30kHz. A discrete-time first-order low-pass filter, also known as an exponentially



(a) (b)

Figure 2.10: Environment for repeatability test. (a) Pose 1, (b) pose 2.

weighted moving average, is involved in the derivative term, and saturation is applied in the integral term. The control cycle of the PID controller is set to be 12kHz.

For high-level control of the assembled manipulator, a master controller is implemented in the Xenomai-based real-time environment. To generate smooth high-level command, we implemented a basic trapezoidal joint space planner. After calculating the joint trajectories, the master controller provides a reference position and velocity to each low-level controller every 1ms.

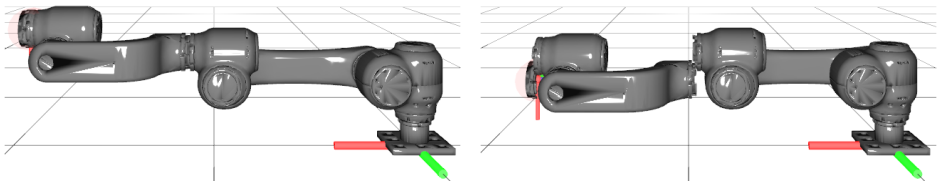
2.3.2 Repeatability Comparison with Non-reconfigurable Robot Manipulators

To evaluate repeatability, a left-arm six-dof configuration manipulator repeated a motion between two postures based on the trajectory planned by the trapezoidal joint space planner. The values of a calibrated dial gauge are recorded when the

Trial	Displacements(mm)	Trial	Displacements(mm)
1	4.960	11	4.960
2	4.965	12	4.960
3	4.960	13	4.960
4	4.960	14	4.960
5	4.965	15	4.965
6	4.965	16	4.950
7	4.970	17	4.950
8	4.960	18	4.960
9	4.970	19	4.950
10	4.970	20	4.950
Stand. Div.	**0.006305**	Max Diff	**0.02**

Table 2.3: Repeatability test results

robot hits the gauge tip with the second posture as illustrated in Fig. 2.10. The result is shown in Table 2.3. The maximum difference among values is $0.02mm$ with a standard deviation of $0.006305mm$. One of the most widely used non-reconfigurable manipulators, UR5 of Universal Robot, has $\pm 0.1mm$ repeatability [48]. The result proves that our manipulator system is indeed comparable to existing non-reconfigurable manipulators in terms of repeatability. Note that the ModMan system only relies on the independent joint control. It implies that by applying a more advanced type of control method, the ModMan system can have higher repeatability.



(a) (b)

Figure 2.11: Illustrations of kinematics for offset comparison. (a) With designed offset, (b) without offset.

2.3.3 Effect of the Offset in Two-dof Modules

To investigate the effect of the offset between the two joints on the safety of the robot, performance indices related to safety were obtained for robots with and without offsets which is illustrated in Fig. 2.11. The robot of Fig. 2.11a is six-dof configuration with designed two-dof modules for ModMan and the robot of Fig. 2.11a is also six-dof configuration with the offsets are removed manually.

Kinematic manipulability of [49] $\mu = \sqrt{\det(JJ^T)} = \sqrt{\sigma_1\sigma_2\sigma_3}$ and the reciprocal of the condition number of [50] $\kappa^{-1} = \sqrt{\frac{\sigma_3}{\sigma_1}}$ are two of the most popular performance measures that represent the degree of independence of the columns of the robot's Jacobian matrix at a certain joint values, where $sigma_i$ is the eigenvalue of JJ^T. Since collision can occur in any direction, condition number to second biggest eigenvalue σ_2, $\kappa_2^{-1} = \sqrt{\frac{\sigma_2}{\sigma_1}}$, is also important for safety. To compare performance between different manipulators, global measures are defined as the

Performance indices	wOffset	woOffset
$\max \mu$	0.1586	0.1308
$\max \kappa^{-1}$	0.8627	0.8251
$\max \kappa_2^{-1}$	1.0000	0.9291
GMI	67.3864	61.5076
GCI	411.9038	405.7943
GCI_2	901.2507	874.5171

Table 2.4: Comparison of performance measures of manipulators with and without offset at two-dof modules.

integrals of μ, κ^{-1}, and κ_2^{-1}, respectively, over entire configuration space as

$$GMI = \int \mu dC \tag{2.3.3}$$

$$GCI = \int \frac{1}{\kappa} dC \tag{2.3.4}$$

$$GCI_2 = \int \frac{1}{\kappa_2} dC \tag{2.3.5}$$

where $dC = dq_1 \cdots dq_n$.

Since q_1 and q_6 does not change J_v, we uniformly sample $q_2, \cdots, q_5$ and evaluate the aforementioned performance indices. The results are shown in Table. 2.4. All the maximum values of the local indices and the global measures are bigger in the configuration with offsets, which the fact implies that the offset between the axes in the two-dof module indeed increases the safety.

2.4 Conclusion

In this chapter, the hardware development of the ModMan is described. It is shown that the proposed genderless connector has sufficient mechanical/electrical capacity to drive an industrial robot. The ModMan system has seven joint modules and four link modules that can be assembled to configure various types of manipulators including widely used six-dof and seven-dof manipulators. Experimental results demonstrate that the repeatability of ModMan is comparable to that of conventional non-reconfigurable manipulators and the offset between the axes in the two-dof modules enhances the safety.

By expanding the kinds of ModMan hardware modules, a user will be able to configure various other types of robot manipulators such as a Selective Compliance Assembly Robot Arm (SCARA) robot or a spherical robot.

3

A Programming Architecture for Modular Reconfigurable Robots

For the development of an easy-to-use software for modular reconfigurable robots, we propose a novel programming architecture. The architecture is divided into a data structure, a communication protocol, and a modeling algorithm. To deal with genderless connectors and multi-dof modules of the ModMan, we define a tree-structured generalized module and construct a data structure that defines the properties of the generalized module. By classifying the connection-dependent parameters, it is possible to express the connection relationship between the modules with a simple communication protocol. We proposed an algorithm that automatically models the kinematics and dynamics of the reconfigurable modular robot by combining the parameters inherent to the module and the parameters that change depending on the connection. Lastly, more accurate control is achieved by deriving a recursive dynamics of robots with flexible joints.

This chapter is organized as follows. Section 3.1 describes a data structure for

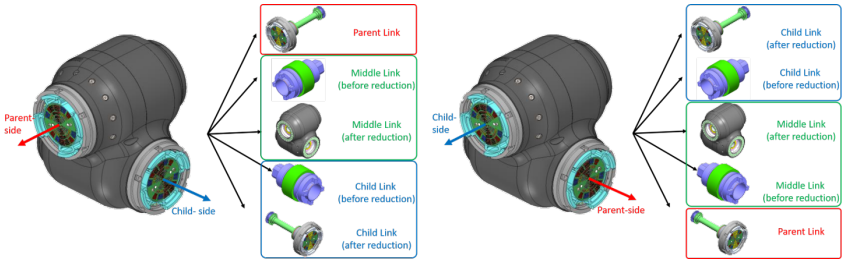
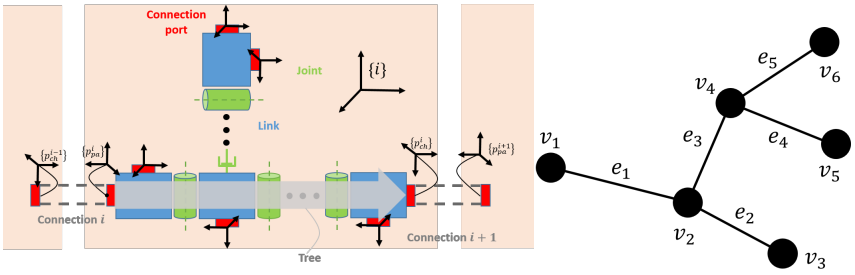


Figure 3.1: Decomposition of inertial components depending on the connection direction.

a generalized tree-structured module. In Section 3.2, an automatic kinematic modeling algorithm with modular kinematics is presented. In Section 3.3, an accurate dynamic parameter initialization method is described with active/inactive states of the joints. The recursive dynamic equation for the robot with flexible joints is also described in Chapter 3.4. Section 3.5 shows experimental results regarding reconfigurability of ModMan. Section 3.6 concludes this chapter with further comments.

3.1 Data Structure for Multi-dof Joint Modules

In this section, we propose a data structure for each module. Due to the genderless connector and multi-dof modules, ModMan's kinematic and dynamic modeling requires additional consideration of how to split joints and links. For example, a two-dof joint module of the minimum number of connectors consists of three link elements, two joint elements, and two connection ports as illustrated in Fig. 3.1. The joint axis parameters and the inertial parameters will vary depending on the connection direction and the connected port. To cope with this problem, we present



(a) (b)

Figure 3.2: Illustration of a multi-dof joint with bidirectional ports. (a) generalized module, (b) graph for joints and links

a new data structure that describes a multi-dof joint module with multiple connection ports.

To consider more general types of modules, we first defined a *generalized module* as illustrated in Fig. 3.2a. The generalized module itself is a tree structure that consists of N_J joints, N_L links, and N_P bidirectional connection ports. To describe the tree structure of the module, we use a incidence matrix defined by the basic graph theory, which is introduced in Appendix A.1. When the links and the joints are represented by vertices and edges, respectively, the incidence matrix X for an example graph shown in Fig 3.2b is as follows.

$$X_I = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}. \tag{3.1.1}$$

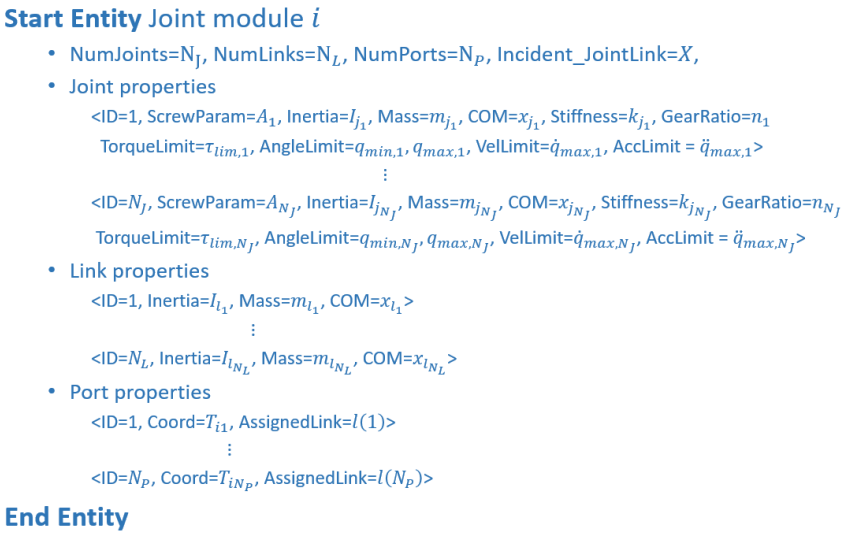


Figure 3.3: Proposed data structure for a generalized module

The properties of this generalized module can be represented in the form of a data structure, as shown in the Fig 3.3. In addition to the number of joints, links, and connection ports and the incidence matrix, the data structure has properties for each component. All coordinate dependent parameters, such as joint screw parameters, joint/link inertias, COM positions, and connection port coordinates, are described in the module frame $\{i\}$. Each joint is assumed to be either a revolute or prismatic joint.

Before a generalized module is connected to other modules, the relationship between joints and links has no directionality. When the module is connected to one parent module and one or several child modules, directional branches are formed inside the module, and the active/inactive state of each joint is determined. If a joint is inactive, that is, it does not belong to the connected branches, we assume that the motor brake of the joint is triggered and the joint is considered as a

link. To represent the state of these joints, we newly defined the *activity matrix* X_A, which leaves the components corresponding to the active joints in the incidence matrix and zeros the rest. In the example of Fig 3.2b, if v_4 is connected to the parent module and v_2 and v_3 are connected to child modules, the joints corresponding to e_2, e_3 are active and the remaining joints are inactive. In this situation, the activity matrix is as follows.

$$X_A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}. \tag{3.1.2}$$

This activity matrix is used in the process of obtaining the dynamic parameters, which will be described later in this chapter.

Suppose that a total of N_G generalized modules are connected to form a tree-type robot with N_T end-effectors. The connection relationship and direction of connected modules can be described by the assembly incidence matrix(AIM) and accessibility matrix(AM), respectively, presented in [51].

3.2 Automatic Kinematic Modeling

In this section, we propose an algorithm that automatically performs kinematic modeling using generalized modules. The main idea of the algorithm is to obtain the kinematics of the entire robot by obtaining the kinematics of each module

and multiplying them. In what follows, we use the modern screw-theoretic formulation of kinematics and dynamics as outlined in [46, 52]. We refer the reader to Appendix A.2 for specific notations.

Let's look at the i'th module belonging to one of the N_T branches. When connected to the parent module and the child module, a unique branch B^i with k^i joints is determined within the module. B^i can be represented as an ordered collection of indices as

$$B^i = \{p^i_{pa}, j(1), \cdots, j(k^i), p^i_{ch}\}, \tag{3.2.3}$$

where p^i_{pa} is the index of parent-side connection port, p^i_{ch} is the index of child-side connection port, and $j(r), r = 1, \cdots, k^i$ are the indices of joints. Let us assume that m modules are serially connected to construct a n-dof manipulator and each module has its own module frame $\{i\}, i = 1, \cdots, m$. The forward kinematics $T_{0,e}$, which is the transformation from base frame $\{0\}$ to end-effector frame $\{e\}$, can be expressed as

$$T_{0,e} = T_{0,p^1_{pa}} T_{p^1_{pa},p^2_{pa}} \cdots T_{p^{m-1}_{pa},p^m_{pa}} T_{p^m_{pa},e} \tag{3.2.4}$$

where $T_{p^i_{pa},p^{i+1}_{pa}} \in SE(3)$ is a *modular kinematics* of module i which is the transformation matrix from parent port frame of module i, $\{p^i_{pa}\}$, to parent port frame of module $i+1$, $\{p^{i+1}_{pa}\}$. $T_{p^m_{pa},e} \in SE(3)$ is the transformation matrix defined at the end-effector module. Assuming all link frames are initially coincide with the module frame $\{i\}$, the transformation from module frame $\{i\}$ to link frame $\{l\}$, $T_{i,l}, l = 0, \cdots, k_i$, can be described as

$$T_{i,l} = \begin{cases} I_{4\times4} & \text{if } l = 0 \\ e^{\left[A^i_{j(1)}\right]q_{\psi+1}} \cdots e^{\left[A^i_{j(l)}\right]q_{\psi+l}} & \text{otherwise,} \end{cases} \tag{3.2.5}$$

where $\psi = \sum_{j=1}^{i-1} k_j$. Now the modular kinematics of module i can be described as

$$T_{p_{pa}^i, p_{pa}^{i+1}} = T_{i,p_{pa}^i}^{-1} e^{\left[A_{j(1)}^i\right] q_{\psi+1}} \cdots e^{\left[A_{j(k^i)}^i\right] q_{\psi+k_i}} T_{i,p_{ch}^i} T_{p_{ch}^i, p_{pa}^{i+1}} \tag{3.2.6}$$

where $T_{i,p_{pa}^i}, T_{i,p_{ch}^i} \in SE(3)$ are initial transformations from module frame $\{i\}$ to parent port frame $\{p_{pa}^i\}$, child port frame $\{p_{ch}^i\}$, respectively, $T_{p_{ch}^i, p_{pa}^{i+1}}$ is the transformation from i'th child port frame $\{p_{ch}^i\}$ to $i+1$'th parent port frame $\{p_{pa}^{i+1}\}$, of which the value is dependent on offset angle ϕ_i between module i and module $i+1$. For systematic derivation of $T_{p_{ch}^i, p_{pa}^{i+1}}$, we assume that the orientation of the port frame is defined in a consistent manner: in this thesis we choose the outward normal direction at a connection port to be $\hat{z}$-direction and the direction of the reference line for the offset angle to be $\hat{x}$-direction. Now $T_{p_{ch}^i, p_{pa}^{i+1}}$ can be defined as

$$T_{p_{ch}^i, p_{pa}^{i+1}}(\phi_i) = \begin{pmatrix} \cos\phi_i & \sin\phi_i & 0 & 0 \\ \sin\phi_i & -\cos\phi_i & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.2.7}$$

Since the intrinsic parameters of the modules are defined in the data structure, the entire kinematics of the branch can be described with connection-dependent parameters. The list of connection-dependent parameters is as follows.

- $B^i = \{p_{pa}^i, j(1), \cdots, j(k^i), p_{ch}^i\}$, the ordered indices for parent port, joints, and child port.
- t^i, the module ID,
- ϕ_i, the offset angle of the connection.

We apply the automatic kinematic modeling algorithm to the ModMan. In the case of the developed two-dof modules, since there are only two branches that can

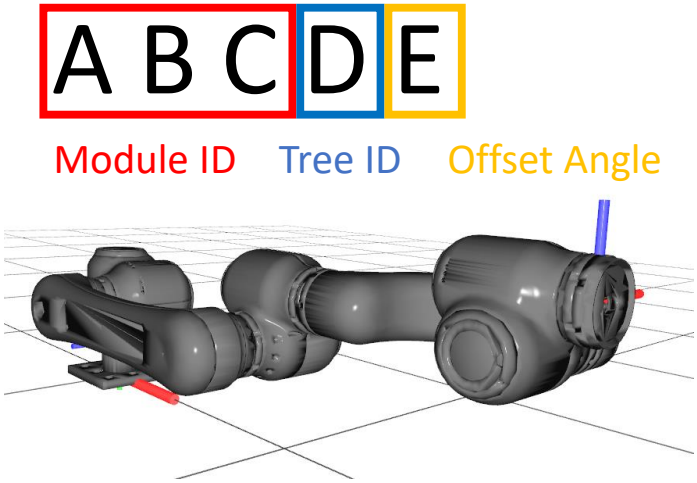


Figure 3.4: Connection protocol and configured manipulator in its zero position with 00010 - 21114 – 00212 – 22214 – 00512 – 23112 – 00101 (Base-LargeA-LinkA-MedB-LinkB-SmallA-Dummy)

be connected, connection-dependent parameters can be represented using a simple protocol, five-digit integer, as shown in Fig. 3.4. The first three digits represent the module ID, fourth and fifth digits represent tree ID and offset angle, respectively. Now the connection status of a configured manipulator can be represented as a series of integers. An example of six-dof right-arm configuration is illustrated in Fig. 3.4.

3.3 Automatic Dynamic Modeling

This section introduces an automatic dynamic modeling algorithm of modular robots. Since the velocity and acceleration of each link frame can be obtained from the

Algorithm 1 Link dynamic parameter estimation

Initialization: $\tilde{G}_0 = \text{given}, \psi_0 = 0$

for $i = 1$ to m **do**

$G_\psi = \tilde{G}_\psi + \left[\text{Ad}_{T_{i,i-1}}\right]^T G^i_{pa} \left[\text{Ad}_{T_{i,i-1}}\right]$

for $l = 1$ to k_i **do**

$G_{\psi+l} = G^i_l$

if $Sum(X^i_{IA}(l,:)) \neq 0$ **then**

$G_{\psi+l} = G_{\psi+l} + \sum_{r\in\lambda^i(l)} G^i_r$

end if

end for

$\tilde{G}_{\psi+k_i} = G_{\psi+k_i}$

$\psi = \psi + k_i$

end for

kinematics obtained in the previous section, the dynamics of the robot can be described by accurately obtaining the inertial parameters assigned to each joint. Since each link frame matches the module frame when $q = 0$, all inertial parameters can be described in the module frame. When a module is connected to the child-side of the current module, the inertia assigned to the last joint of the current module changes, so parameter values must be obtained through forward recursion starting from the base.

An additional consideration in the process of obtaining inertial parameters is the effect of inactive joints and their adjacent links, which are considered as links attached to the current link. By subtracting the activity matrix X^i_A from the module's incidence matrix X^i_I, we can define the inactivity matrix X^i_{IA}, where the components related to the inactive joint are one and the rest are zero. If there

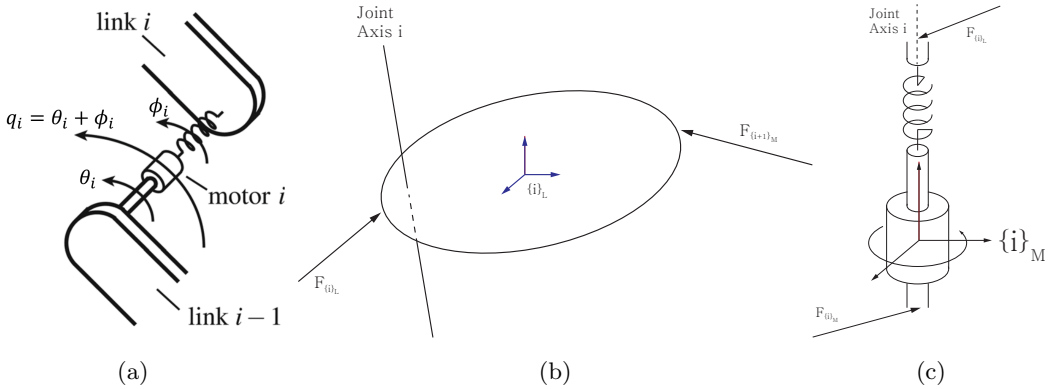


Figure 3.5: Illustrations of of the robot with flexible joint. (a) joint with flexibility, (b) i'th link, (c) i'th joint.

are nonzero components in the row of X_{IA}^i corresponding to the current link, one or more inactive joints exist. If we define a set $\lambda^i(l)$ representing the indices of the inactive joints and their child elements, the summation of the inertias of the elements corresponding to $\lambda^i(l)$ and the inertia of the current link is the final apparent inertia.

Considering the above, a recursive algorithm to obtain the link dynamic parameters is presented in Algorithm 1. The dynamic parameter for the rotor is explicitly considered using the Newton-Euler inverse dynamics algorithm for motor inertias and gearing, outlined in section 8.9.3 of [46].

3.4 Flexibility in Manipulator

Previously in the hardware design of the joint modules, flexible parts are placed in the joints to increase safety. Flexibility in the joints helps to increase safety, but it is a factor that can degrade accuracy in terms of robot control. For precise control,

we derive the dynamics of a robot with flexible joints. For clarity, the notations in this section are used independently of the notations previously defined for modular robots.

Figure 3.5 shows a schematic diagram of a robot with flexible components. Here, q_i, θ_i, and K_i represent the link angle, the motor angle, and the stiffness of i'th joint, respectively. Given i'th link frame i_L and motor frame i_M, what is described below explains the notations used in the dynamics equation.

The generalized velocity $\mathcal{V}_{i_L}$ of the i'th link is the same as that of the rigid body, which is

$$\mathcal{V}_{i_L} = [\mathrm{Ad}_{T_{i_L,i-1_L}}]\mathcal{V}_{i-1_L} + \mathcal{A}_{i_L}\dot{q}_i. \tag{3.4.8}$$

As you can see in Fig 3.5b, the force applied to the i'th link is the sum of the force applied from the spring $\mathcal{F}_{i_L}$ and the force applied from the next motor $\mathcal{F}_{i+1_M}$, and the equation of motion is

$$G_{i_L}\dot{\mathcal{V}}_{i_L} - [\mathrm{ad}_{\mathcal{V}_{i_L}}]^T G_{i_L}\mathcal{V}_{i_L} = \mathcal{F}_{i_L} - [\mathrm{Ad}_{T_{i+1_M,i_L}}]^T\mathcal{F}_{i+1_M}. \tag{3.4.9}$$

The torque applied to the i'th spring is simply the projection of the link force $\mathcal{F}_{i_L}$ to screw $\mathcal{A}_{i_L}$. From the equation of spring, the following equation is satisfied.

$$K_i(q_i - \theta_i) = \mathcal{F}_{i_L}^{\mathrm{T}}\mathcal{A}_{i_L}. \tag{3.4.10}$$

In similar manner, the generalized velocity of the motor is

$$\mathcal{V}_{i_M} = [\mathrm{Ad}_{T_{i_M,i-1_L}}]\mathcal{V}_{i-1_L} + \mathcal{A}_{i_M}\dot{\theta}_i. \tag{3.4.11}$$

According to [53], using this velocity as it is, the resulting differential equation becomes infinite order. When the reduction gear ratio is large, we can assume that $\mathcal{A}_{i_M}\dot{\theta}_i$ is dominant in $\mathcal{V}_{i_M}$. Then the simplified velocity is

$$\mathcal{V}_{i_M} = \mathcal{A}_{i_M}\dot{\theta}_i. \tag{3.4.12}$$

The dynamics of motor i can be written as

$$B_{i_M}\dot{\mathcal{V}}_{i_M} - [\mathrm{ad}_{\mathcal{V}_{i_M}}]^T B_{i_M}\mathcal{V}_{i_M} = \mathcal{F}_{i_M} - [\mathrm{Ad}_{T_{i_L,i_M}}]^T \mathcal{F}_{i_L}. \tag{3.4.13}$$

The motor torque τ_i is also a projection of the motor force $\mathcal{F}_{i_M}^{\mathrm{T}}$ to screw $\mathcal{A}_{i_M}$, which is given as

$$\tau_i = \mathcal{F}_{i_M}^{\mathrm{T}} \mathcal{A}_{i_M} \tag{3.4.14}$$

The resulting dynamic equation is the fourth-order differential equation and include $\dot{\mathcal{F}}_{i_L}$ and $\ddot{\mathcal{F}}_{i_L}$. The Recursive Newton-Euler inverse dynamics algorithm is summarized as follows.

Assumption : The kinetic energy of the rotors is due only to their own spinning.

(i) **Initialization** : $\mathcal{V}_{0_L}$, $\dot{\mathcal{V}}_{0_L}$, $\mathcal{V}_{0_L}{}^{(2)}$, $\mathcal{V}_{0_L}{}^{(3)}$ = given, $\mathcal{F}_{n+1_M} = \mathcal{F}_{tip}$

(ii) **Forward Iteration** : for $i = 1$ to i do

$$\begin{aligned}
T_{i-1_L,i_L} &= M_{i-1_L,i_L} e^{[\mathcal{A}_{i_L}]q_i} \\
\mathcal{V}_{i_L} &= [\mathrm{Ad}_{T_{i_L,i-1_L}}]\mathcal{V}_{i-1_L} + \mathcal{A}_{i_L}\dot{q}_i \\
\dot{\mathcal{V}}_{i_L} &= [\mathrm{Ad}_{T_{i_L,i-1_L}}]\dot{\mathcal{V}}_{i-1_L} + [\mathcal{V}_{i_L},\mathcal{A}_{i_L}]\dot{q}_i + \mathcal{A}_{i_L}\ddot{q}_i \\
\mathcal{V}_{i_L}{}^{(2)} &= [\mathrm{Ad}_{T_{i_L,i-1_L}}]\mathcal{V}_{i-1_L}{}^{(2)} + [\mathcal{A}_{i_L},[\mathcal{V}_{i_L},\mathcal{A}_{i_L}]]\dot{q}_i^2 + 2[\dot{\mathcal{V}}_{i_L},\mathcal{A}_{i_L}]\dot{q}_i \\
&\quad +[\mathcal{V}_{i_L},\mathcal{A}_{i_L}]\ddot{q}_i + \mathcal{A}_{i_L} q_i{}^{(3)} \\
\mathcal{V}_{i_L}{}^{(3)} &= [\mathrm{Ad}_{T_{i_L,i-1_L}}]\mathcal{V}_{i-1_L}{}^{(3)} + [\mathcal{A}_{i_L},[\mathcal{A}_{i_L},[\mathcal{V}_{i_L},\mathcal{A}_{i_L}]]]\dot{q}_i^3 \\
&\quad +3[\mathcal{A}_{i_L},[\dot{\mathcal{V}}_{i_L},\mathcal{A}_{i_L}]]\dot{q}_i^2 + 3[\mathcal{A}_{i_L},[\mathcal{V}_{i_L},\mathcal{A}_{i_L}]]\ddot{q}_i\dot{q}_i + 3[\dot{\mathcal{V}}_{i_L},\mathcal{A}_{i_L}]\ddot{q}_i \\
&\quad +3[\ddot{\mathcal{V}}_{i_L},\mathcal{A}_{i_L}]\dot{q}_i + [\mathcal{V}_{i_L},\mathcal{A}_{i_L}] q_i{}^{(3)} + \mathcal{A}_{i_L} q_i{}^{(4)}
\end{aligned}$$

(iii) **Backward Iteration** : : for $i = n$ to i do

Step 1 Force applied to link and motor angle, $\mathcal{F}_{i_L}$ and θ_i

$$\begin{aligned}
\mathcal{F}_{i_L} &= [\mathrm{Ad}_{T_{i+1_L,i_L}}]^T \mathcal{F}_{i+1_L} + G_{i_L}\dot{\mathcal{V}}_{i_L} - [\mathrm{ad}_{\mathcal{V}_{i_L}}]^T G_{i_L}\mathcal{V}_{i_L} \\
\dot{\mathcal{F}}_{i_L} &= G_{i_L}\ddot{\mathcal{V}}_{i_L} - [\mathrm{ad}_{\dot{\mathcal{V}}_{i_L}}]^{\mathrm{T}} G_{i_L}\mathcal{V}_{i_L} - [\mathrm{ad}_{\mathcal{V}_{i_L}}]^{\mathrm{T}} G_{i_L}\dot{\mathcal{V}}_{i_L} + [\mathrm{Ad}_{T_{i+1_L,i_L}}]^{\mathrm{T}}\dot{\mathcal{F}}_{i+1_L} \\
&\quad + [\mathrm{ad}_{A_{i+1_L}}]^T [\mathrm{Ad}_{T_{i+1_L,i_L}}]^{\mathrm{T}} \mathcal{F}_{i+1_L}\dot{q}_{i+1} \\
\ddot{\mathcal{F}}_{i_L} &= G_{i_L}{\mathcal{V}_{i_L}}^{(3)} - [\mathrm{ad}_{\ddot{\mathcal{V}}_{i_L}}]^{\mathrm{T}} G_{i_L}\mathcal{V}_{i_L} - 2[\mathrm{ad}_{\dot{\mathcal{V}}_{i_L}}]^{\mathrm{T}} G_{i_L}\dot{\mathcal{V}}_{i_L} - [\mathrm{ad}_{\mathcal{V}_{i_L}}]^{\mathrm{T}} G_{i_L}\ddot{\mathcal{V}}_{i_L} \\
&\quad + [\mathrm{Ad}_{T_{i+1_L,i_L}}]^{\mathrm{T}}\ddot{\mathcal{F}}_{i+1_L} + 2[\mathrm{ad}_{A_{i+1_L}}]^T [\mathrm{Ad}_{T_{i+1_L,i_L}}]^{\mathrm{T}}\dot{\mathcal{F}}_{i+1_L}\dot{q}_{i+1} \\
&\quad + [\mathrm{ad}_{A_{i+1_L}}]^T [\mathrm{Ad}_{T_{i+1_L,i_L}}]^{\mathrm{T}} \mathcal{F}_{i+1_L}\ddot{q}_{i+1} \\
&\quad + [\mathrm{ad}_{A_{i+1_L}}]^T [\mathrm{ad}_{A_{i+1_L}}]^T [\mathrm{Ad}_{T_{i+1_L,i_L}}]^{\mathrm{T}} \mathcal{F}_{i+1_L}\dot{q}_{i+1}^2 \\
\theta_i &= q_i + {K_i}^{-1}\mathcal{A}_{i_L}^{\mathrm{T}}\mathcal{F}_{i_L}
\end{aligned}$$

Step 2 Motor velocity and motor acceleration, $\dot{\theta}_i$ and $\ddot{\theta}_i$

$$\begin{aligned}
\dot{\theta}_i &= \dot{q}_i + {K_i}^{-1}{\mathcal{A}_{i_L}}^T\dot{\mathcal{F}}_{i_L} \\
\ddot{\theta}_i &= \ddot{q}_i + {K_i}^{-1}{\mathcal{A}_{i_L}}^T\ddot{\mathcal{F}}_{i_L}
\end{aligned}$$

Step 3 Motor generalized velocity and acceleration, $\mathcal{V}_{i_M}$ and $\dot{\mathcal{V}}_{i_M}$

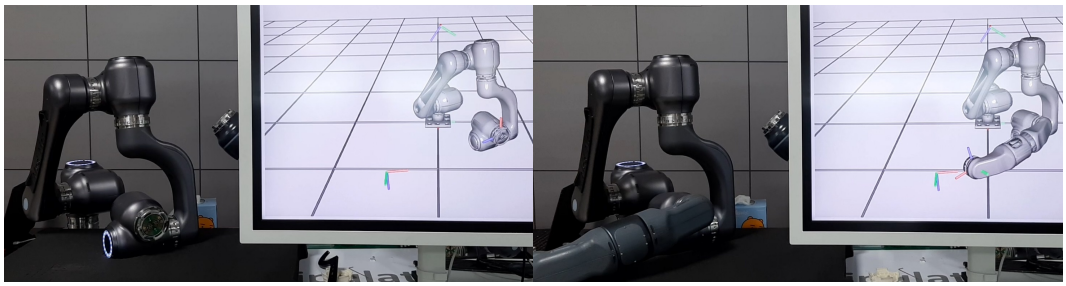
$$\begin{aligned}
\mathcal{V}_{i_M} &= \mathcal{A}_{i_M}\dot{\theta}_i \\
\dot{\mathcal{V}}_{i_M} &= \mathcal{A}_{i_M}\ddot{\theta}_i
\end{aligned}$$

Step 4 Force applied to motor and motor torque, $\mathcal{F}_{i_M}$ and τ_i

$$\begin{aligned}
\mathcal{F}_{i_M} &= B_{i_M}\dot{\mathcal{V}}_{i_M} \\
\tau_i &= \mathcal{A}_{i_M}^T\mathcal{F}_{i_M} + K_i(\theta_i - q_i)
\end{aligned}$$

3.5 Experiments

We performed two different hardware experiments. Reconstruction tests are also conducted to verify that the automatic kinematic modeling algorithm presented in



(a) (b)

Figure 3.6: Automatic kinematic modeling test. (a) six-dof, (b) seven-dof.

the previous section works well.

3.5.1 System Setup

To show that the automatic kinematic modeling algorithm works on actual hardware, a software to visualize the ModMan system is built. When all the modules are connected, all the slaves send their connection-dependent parameters to the master controller, which are forwarded to the visualization software. The kinematics of the manipulator is constructed using the algorithm of Section 3.2 and the simulation model is synchronized with the actual hardware using joint encoder values which is received every 10ms through TCP/IP communication. 3D mesh models of the modules are separated into each link as illustrated in Fig. 3.1 and all link coordinates are computed using (3.2.5).

3.5.2 Reconfigurability

Two different combinations of modules are tested for real-time kinematic modeling and the results are shown in Fig. 3.6. The first configuration is a six-dof right-arm

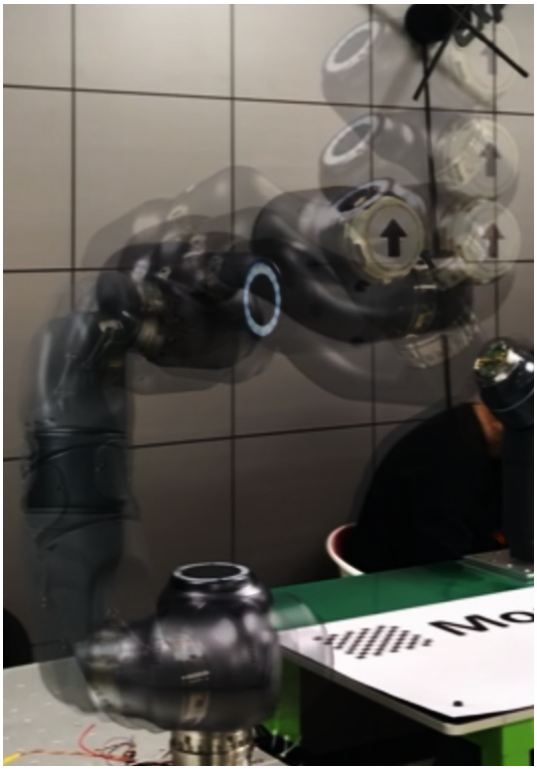


Figure 3.7: Time-lapse photograph of Cartesian motion seven-dof configuration

manipulator, which resembles the properties of conventional elbow-type manipulators. The second configuration is formed by connecting a one-dof module at the end of the first configuration. The automatic kinematic modeling algorithm constructs the kinematics of the two manipulators successfully; the postures of the simulation models are identical to the postures of the actual hardware.

To confirm that the Jacobian of the robot is properly found at every instance, the robot is to follow the trajectory defined in the task space. Manipulator Jacobian for differential kinematics can easily be obtained by transforming the reference coordinate of screw parameters of each joint to the end-effector frame. The

(a) (b)

Figure 3.8: A pick-and-place task with iPad. (a) position estimation of an object, (b) picking motion.

Newton-Raphson numerical inverse kinematics described in [46] is implemented, since the algorithm is adaptable for manipulators of arbitrary kinematics. A Cartesian space planner is implemented using the numerical inverse kinematics and B-spline interpolation.

A seven-dof configuration is subjected to a task space trajectory whose orientation is constant and the result is illustrated in Fig. 3.7. The inverse of the Jacobian matrix $J \in \mathbb{R}^{6\times7}$ is calculated with Moore-Penrose pseudo inverse for minimum norm solution. The end-effector successfully keeps its orientation constant during the motion.

3.5.3 Pick-and-Place with Vision Sensors

To show the ModMan can be integrated to vision systems, we conducted pick-and-place tasks using vision sensors as illustrated in Fig. 3.8. Here we use an RGB camera on iPad and an RGB-D camera mounted on the end-effector module. First,

we create a shared coordinate system by shooting one common chessboard from two cameras. By using an object pose estimation algorithm with an RGB image introduced in [54], we can find the position of the target object from the iPad camera by shooting the object. Since the iPad can localize itself through its own simultaneous localization and mapping (SLAM) algorithm called ARKit [55], the position of the object from iPad camera can be transformed to the position from the robot base. Fig. 3.8b shows that the robot successfully picks the target object.

3.6 Conclusion

In this chapter, we present a programming architecture for modular reconfigurable manipulator systems. To deal with more general types of modules, we define a generalized module and a data structure to represent the properties of the module. By distinguishing connection-dependent parameters from module-intrinsic parameters, it is possible to obtain the kinematics and dynamics of the assembled robot with a simple protocol. Using the proposed recursive dynamics algorithm for robot with flexible joints, model errors caused by adding a torsion bar can be reduced. The reconfigurability of ModMan is validated with the real hardware with multiple arbitrary combinations of the modules. A set of experiments that utilizing constructed kinematics are also performed.

If the mod man has a torque control mode, the automatic dynamic modeling algorithm allows us to come up with more sophisticated torque control algorithms and allows the ModMan to be reactive to external forces for safety.

4

A Preparatory Safety Measure for Robust Collision Avoidance

In this chapter, we describe a collision avoidance algorithm that is robust to the kinematic singularity of serial robots. The algorithm is combining *repulsive action*, generated by conventional Cartesian space potential functions of [18], and *preparatory action*, which increases the directional manipulability along a direction of repulsive action applied. A hierarchical velocity control framework is used to incorporate with a nominal task. Real-time computation of the control input is realized by utilizing the Lie-theoretic formulation of robot kinematics and dynamics.

The chapter is organized as follows. In Section 4.1, static and dynamic viewpoints related to the general safety of robots are described. Section 4.2 describes the analysis of the reflected mass and modifies the formulation to alter its unwanted properties. The geometric properties of the space of positive semi-definite

matrices and the solution to the rank-1 manipulability set-point problem are described in Section 4.3. The collision avoidance algorithm with hierarchical velocity control is given in Section 4.4. A set of experiments for manipulability set-point problem and collision avoidance is demonstrated in 4.5. The conclusion is presented in Section 4.6.

4.1 Preliminaries on Manipulability and Safety

In this section, we briefly review the relationship between the manipulability and the safety of a serial robot. We refer the readers to [28] for more details on safety and manipulability.

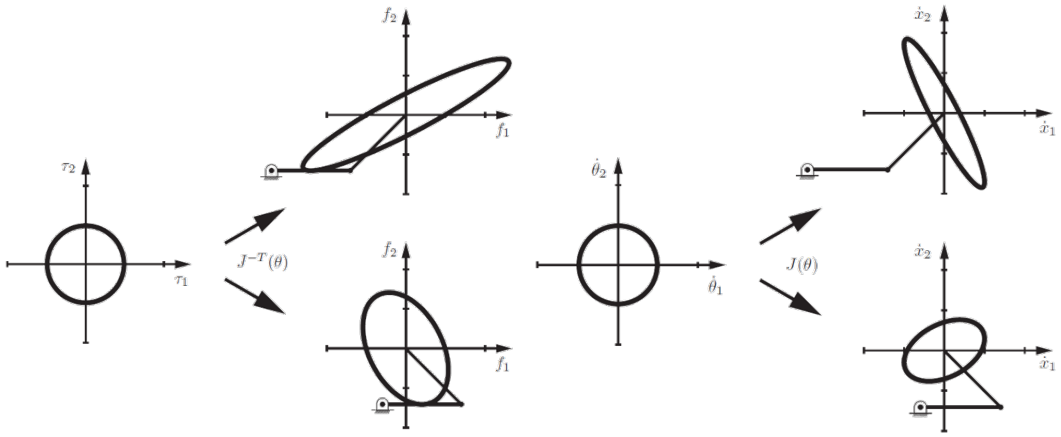
When a collision occurs between an object and a robot, a force applied to an object is equally transmitted to the robot by the Newton's third law of motion. Therefore, the safety of the object can be known by analyzing how the robot responds to external shocks. First we examine a simple example of a stationary robot. The relationship between F_{ext}^{c} and τ_{ext} is described as

$$\tau_{ext} = J^{cT} F^{c}. \tag{4.1.1}$$

Unit torque condition generates an ellipsoid as

$$1 = \tau^{T}\tau = F^{cT} J^{c} J^{cT} F^{c}. \tag{4.1.2}$$

Equation (4.1.2) represent a well-known force ellipsoid as illustrated in Figure 4.1a. In terms of generating force, the direction with bigger length in the ellipsoid means the robot can exert bigger force to the environment with the same amount of torque. The same situation means the object can be damaged more severely with bigger force. If $J^{c}J^{cT}$ becomes singular, theoretically the magnitude



(a) (b)

Figure 4.1: Manipulability ellipsoids. (a) force ellipsoid, (b) velocity ellipsoid

of the force can diverge to infinity which should be avoided in human-robot interaction. So in terms of safety, the direction of force should be close to the minor axis of the force ellipsoid.

On the other hand, an ellipsoid from unit joint velocity condition is

$$1 = \dot{q}^T \dot{q} = V^{cT} (J^c J^{cT})^{-1} V^c. \tag{4.1.3}$$

The relationship between force and velocity ellipsoid is that the principal axes have the same direction, and the magnitudes are reciprocal. So the control objective for safety is to make the direction of interest close to the minor axis of the force ellipsoid or, equivalently, the major axis of the velocity ellipsoid.

The safety of the robot can also be determined through dynamic analysis. With the assumption of infinitesimal time of acting as described in [28,31], the impulsive dynamics of the robot produces the change in velocity of the robot at the control

point {c}, which is

$$\Delta v^c_{robot} = J^c_v(q)M(q)^{-1}J^c_v(q)^T p = K_{robot}(q)p, \tag{4.1.4}$$

where $p \in \mathbb{R}^3$ is the impulse exerted on the robot in the direction of the change in relative velocity $v^c_{rel} = v^c_{robot} - v^c_{obj}$, described in {c}. $K_{robot}(q) \in \mathbb{R}^{3\times3}$ is the translation part of the inverse of Cartesian mass matrix at frame $\{c\}$, which is a symmetric positive (semi-)definite matrix. The change in velocity of the object also is of similar form, as described in [56]:

$$\Delta v^c_{obj} = \left(\frac{1}{m_{obj}} I_{3\times3} + \left[r^c_{obj}\right]^T G_{obj}[r^c_{obj}]\right)(-p) = -K_{obj}p \tag{4.1.5}$$

where $m_{obj} \in \mathbb{R}$ is the mass of the object, $r^c_{obj} \in \mathbb{R}^3$ is the position vector of the collision point from the object center as shown in Fig. 1.2. $G_{obj} \in \mathbb{R}^{3\times3}$ is the symmetric positive-definite rotational inertia of the object. Assuming the object is rigid and spherical shape, the values of m_{obj}, G_{obj}, and r_{obj} are constant, so that $K_{obj} \in \mathbb{R}^{3\times3}$ is also constant and positive-definite.

If we subtract (4.1.5) from (4.1.4) and use the relationship $p = \frac{\|p\|}{\|v^c_{rel}\|} v^c_{rel}$ and Newton's impact law $\Delta v^c_{rel} = (1+e)v^c_{rel}$, the impulse can be represented as

$$\|p\| = \frac{(1+e)\|v^c_{rel}\|}{{u^c_{rel}}^T \left(K_{robot}\left(q\right) + K_{obj}\right) u^c_{rel}} = \frac{1+e}{\mu_{robot} + \mu_{obj}} \|v^c_{rel}\|, \tag{4.1.6}$$

where

$$\mu_{robot}(q) = {u^c_{rel}}^T K_{robot}\left(q\right) u^c_{rel} \tag{4.1.7}$$

$$\mu_{obj} = {u^c_{rel}}^T K_{obj} u^c_{rel} \tag{4.1.8}$$

are *the inverse of reflected mass* of the robot and the object, respectively, along the unit direction vector $u^c_{rel} = \frac{1}{\|v^c_{rel}\|} v^c_{rel}$, $e \in \mathbb{R}$ is the coefficient of restitution, $\|\cdot\|$ is the standard Euclidean norm in Cartesian space. For same relative velocity and

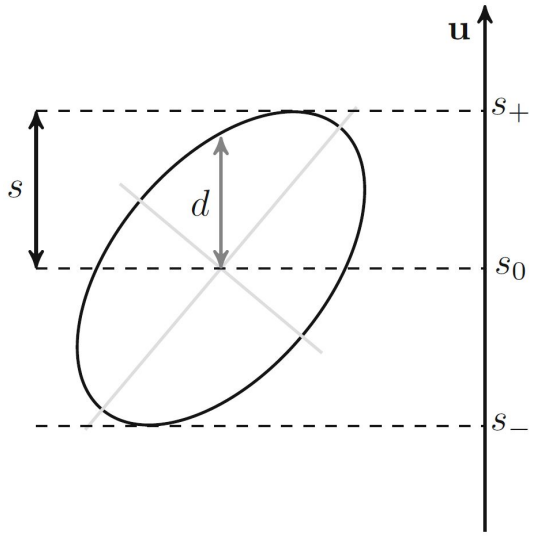


Figure 4.2: An ellipsoid and length measures

constant reflected mass of object, the impulse $\|p\|$ increases as the reflected mass of the robot increases or, equivalently, μ_{robot} decreases. So the control objective becomes increasing μ_{robot}.

To see what μ_{robot} geometrically means, directional quantities of an ellipsoid is examined as described in [57] and illustrated in Fig. 4.2. There are two types of directional measures, the length d of the manipulability ellipsoid in the desired direction and the orthogonal projection s. When the ellipsoid is defined by $x^T A^{-1} x = 1$, the values of d and s, for a vector u with unit length, are given as

$$d = \frac{1}{\sqrt{u^T A^{-1} u}} \tag{4.1.9}$$

$$s = \sqrt{u^T A u}. \tag{4.1.10}$$

So the inverse of the reflected mass μ_{robot} is the square of the orthogonal projection

of the dynamic manipulability ellipsoid which is defined as

$$v^{cT}(J_v^c M^{-1} J_v^{cT})^{-1} v^c = 1. \tag{4.1.11}$$

So the problem of decreasing the reflected mass along u^c is equivalent to increasing the orthogonal projection of the dynamic manipulability along same direction, which agrees with the analysis of kinematic case. For the rest of thesis, we will call the orthogonal projection of any ellipsoid as the directional manipulability.

A more general form of manipulability ellipsoid can be considered to further generalize the results from static and dynamic analysis. We can define a ellipsoid with a square matrix $A_{\text{robot}} \in \mathbb{R}^{m \times m}$ of the form

$$A_{\text{robot}}(q) = \Phi J^c(q) W(q) J^{cT}(q) \Phi^T \tag{4.1.12}$$

where $\Phi \in \mathbb{R}^{m \times 6}$ is the selection matrix and $W(q) \in P(n)$ is the weight matrix. We can set $\Phi = I_{n \times n}$ for full-Jacobian case and $\Phi = (0_{3\times 3} \quad I_{3\times 3})$ for translational manipulability. The type of the manipulability is determined by choosing $W(q)$; $W = I_{n\times n}$ for kinematic manipulability, $W = M^{-1}(q)$ for dynamic manipulability. For any manipulability ellipsoid, the orthogonal projection or its square can be a measure of safety.

4.2 Analysis on Reflected Mass

To utilize the reflected mass for safe control of robots, its gradient with respect to joint angle needs to be calculated. With (4.1.7), we define a safety measure

$$\mu = u^c(q)^T K_{\text{robot}}(q) u^c(q). \tag{4.2.13}$$

where $u^c \in \mathbb{R}^3$ is unit directional vector described in control point frame $\{c\}$, satisfying $u^{cT} u^c = 1$. From (4.1.4), $K_{\text{robot}}(q) \in \mathbb{R}^{3\times 3}$ is symmetric positive-definite

and therefore its eigen decomposition gives

$$K(q) = Q\Lambda Q^T \tag{4.2.14}$$

$$\Lambda = diag\{\lambda_1, \lambda_2, \lambda_2\} \in \mathbb{R}^{3\times 3} \tag{4.2.15}$$

$$Q = \{u_1|u_2|u_3\} \in SO(3), \tag{4.2.16}$$

where eigenvalues satisfying $\lambda_1 \geq \lambda_2 \geq \lambda_3 > 0$ and $u_i, i = 1, 2, 3$ are the eigenvectors of corresponding eigenvalues. The derivatives of (4.2.13) with respect to joint angle are given as follows

$$\frac{\partial \mu}{\partial q_i} = 2u^{cT} Q\Lambda Q^T \frac{\partial u^c}{\partial q_i} + 2u^{cT} \frac{\partial Q}{\partial q_i} \Lambda Q^T u^c + u^{cT} Q \frac{\partial \Lambda}{\partial q_i} Q^T u^c. \tag{4.2.17}$$

Here we see the special cases, when u^c corresponds to j'th eigenvector direction, that is, $u^c = u_j$. With the orthogonality of eigenvectors, (4.2.17) becomes

$$\frac{\partial \mu}{\partial q_i} = \frac{\partial \lambda_j}{\partial q_i} \tag{4.2.18}$$

In the worst case scenario illustrated in Fig 1.2, $j = 3$ and the value of $\frac{\partial \lambda_j}{\partial q_i}$ becomes zero. The value of $\frac{\partial \lambda_3}{\partial q_i}$ is usually small even if the robot is not at the kinematic singularity. This means that when an object comes in a dangerous direction, the robot cannot generate control inputs to escape the dangerous situation.

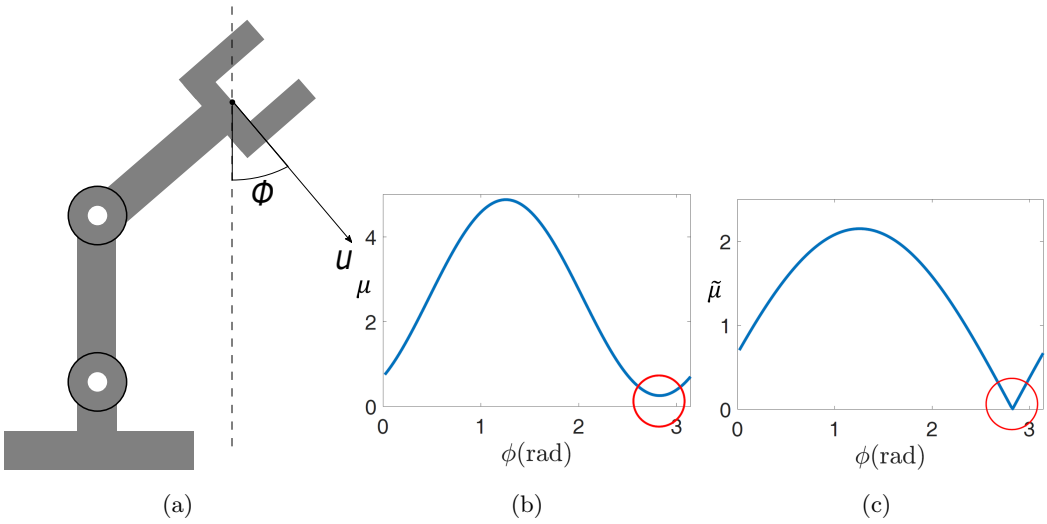
The issue can be resolved by introducing modified reflected mass of the form

$$\tilde{\mu} = \sqrt{\mu - \lambda_3}. \tag{4.2.19}$$

The derivatives of $\tilde{\mu}$ with respect to q and $\dot{q}$ are written as

$$\frac{\partial \tilde{\mu}}{\partial q} = \frac{1}{2\tilde{\mu}} \left(\frac{\partial \mu}{\partial q} - \frac{\partial \lambda_3}{\partial q} \right). \tag{4.2.20}$$

When $u^c = u_3$, the value of $\frac{\partial \tilde{\mu}}{\partial q}$ becomes an indefinite form, $\frac{0}{0}$, and there is a possibility of having a non-zero value.



(a) (b) (c)

Figure 4.3: Comparison of μ and $\tilde{\mu}$ in two-dof planar open-chain arm.

Fig. 4.3 illustrates how the value of μ and $\tilde{\mu}$ at the end-effector of a planar two-dof manipulator varies with $\phi \in \mathbb{R}$, the angle between x-axis of end-effector frame $\{c\}$ and a unit length direction vector $u^c \in \mathbb{R}^3$. As shown in Fig. 4.3b, μ has a zero derivative with respect to ϕ at its local minima. By the chain-rule, $\frac{\partial \mu}{\partial q_k} = \frac{\partial \mu}{\partial \phi}\frac{\partial \phi}{\partial q_k}$, the gradient with respect to joint variable is also zero since $\frac{\partial \phi}{\partial q_k}$ is finite. In contrast, $\frac{\partial \mu}{\partial \phi}$ has a non-zero peak at the local minimum.

Since the position of the minimum and maximum values do not change when a function is composed with a monotonically increasing scalar function, we can consider a more general form of function composition that includes a square root as the following proposition.

Proposition 4.1. *For a monotonically increasing scalar function $g(x)$ of the form*

$$g(x) = \alpha(x - \lambda_3)^\gamma + \beta, \tag{4.2.21}$$

the composition $g \circ \mu(q)$ has non-zero gradient with respect to joint angle q when

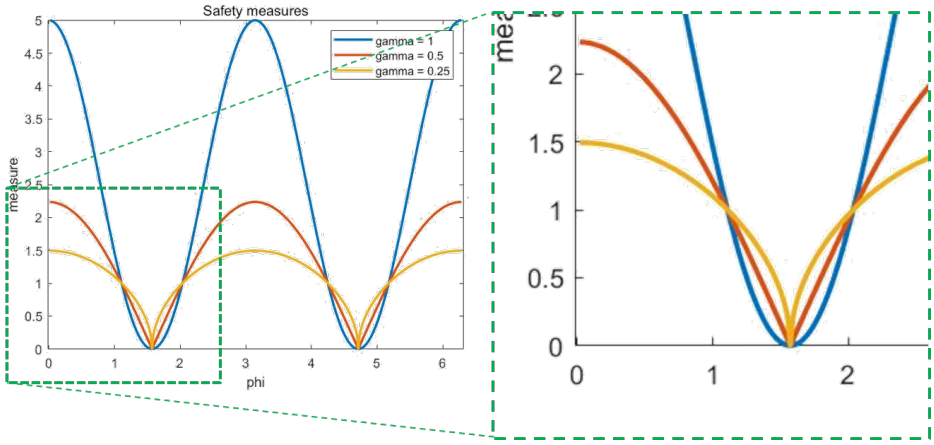


Figure 4.4: The values of (4.2.21) with respect to ϕ

$0 < \gamma \leq 0.5$, $\alpha > 0$, $-\infty < \beta < \infty$. *If* $\gamma = 0.5$, *the composition has non-zero, finite gradient.*

Proof. The proof is provided in Appendix A.4. □

Fig 4.4 shows how the values of (4.2.21) vary for different γ with respect to ϕ. When $0 < \gamma < 0.5$, the gradient value is non-zero but diverges to infinity at the local minimum, which leads to numerical instability. It can be considered that it is best when $\gamma = 0.5$, having a finite gradient value at the local minimum. Therefore, we use $\tilde{\mu}$ of (4.2.19) as a safety measure to send the robot to a safer configuration.

The gradient of (4.2.19) with respect to q is given as

$$\frac{\partial \tilde{\mu}}{\partial q_k} = -\frac{1}{2\sqrt{\mu - \lambda_3}} \left(\frac{\partial \mu}{\partial q_k} - \frac{\partial \lambda_3}{\partial q_k} \right) \tag{4.2.22}$$

and

$$\frac{\partial \mu}{\partial q_k} = 2u^{cT} J_v^c M^{-1} J_v^{cT} \frac{\partial u^c}{\partial q_k} + u^{cT} \left(2 \frac{\partial J_v^c}{\partial q_k} M^{-1} J_v^{cT} + J_v^c \frac{\partial M^{-1}}{\partial q_k} J_v^{cT} \right) u^c. \tag{4.2.23}$$

For our application, analytic gradients of the function are essential for numerical stability, especially in the vicinity of local minima whose gradients are not continuous. Analytic formulations of $\frac{\partial J^c}{\partial q_k}$ and $\frac{\partial M}{\partial q_k}$ can be found in Appendix A.2 and $\frac{\partial \lambda_3}{\partial q_k}$ can be found in Appendix A.3.

It is worth noting that the derivatives of the mass matrix are just a repetition of the derivatives of the screw parameters, so it is possible to be efficiently computed and adaptive to robots with arbitrary kinematics by its recursive formulations. Observe $\frac{\partial u^c}{\partial q_k}$ depends on the choice of u_c; we discuss this case further later.

4.3 Manipulability Control on $S^+(1, m)$

The second approach for defining a new safety measure is using the manipulability matrix itself instead of using its orthogonal projection. If we have a target manipulability matrix $A_{\text{tar}} \in \mathbb{R}^{m\times m}$, the manipulability tracking problem is done by defining and minimizing the distance between $A_{\text{robot}}(q)$ and A_{tar}. If the robot is away from the kinematic singularities, A_{robot} and A_{tar} are full rank and we can use a natural distance on $P(m)$ for tracking. The distance on $P(m)$, however, can be very sensitive near kinematic singularity because it contains matrix inversions. In this section, we propose a manipulability control method on $S^+(1, m)$ that is robust to kinematic singularities.

4.3.1 Geometry of the Group of Positive Semi-definite Matrices

We briefly introduce the geometry of $S^+(p, n)$ presented in [42], where one can find more details. Consider a matrix $A \in S^+(p, n)$, which can always be decomposed

as

$$A = ZZ^T = (U\Sigma)(U\Sigma)^T = U\Sigma^2U^T \tag{4.3.24}$$

where $Z \in \mathbb{R}^{n\times p}$ with rank p, $\Sigma \in P(p)$, and $U \in St(p,n)$. Here $St(p,n)$ is the Stiefel manifold, the set of $n \times p$ matrices with orthonormal columns. For a $R_p \in O(p)$, where $O(p)$ is the orthogonal group, right multiplication of Z by R_p does not change the product (4.3.24), i.e., $ZR_pR_p^TZ^T = ZZ^T$. Thus $S^+(p,n)$ can be represented as a product space $(U,\Sigma) \in St(p,n) \times P(p)$ while it is invariant under $O(p)$ group action. Admitting a quotient manifold representation, $S^+(p,n)$ reduces to

$$\begin{aligned} S^+(p,n) &= (St(p,n) \times P(p))/O(p) &(4.3.25)\\ &= Gr(p,n) \times P(p), &(4.3.26)\end{aligned}$$

where $Gr(p,n)$ is the Grassman manifold, the set of p-dimensional subspaces of $\mathbb{R}^n$.

Let us first focus on $Gr(p,n)$ and $P(p)$ separately and move on to the product space $S^+(p,n)$. From [58], the horizontal tangent space of $Gr(p,n)$ at $U \in Gr(p,n)$ is defined as

$$\begin{aligned} T_UGr(p,n) =&\{U_\perp K | U_\perp \in St(n-p,n),\\ &U^TU_\perp = 0_{p\times(n-p)}, K \in \mathbb{R}^{(n-p)\times p}\}\end{aligned} \tag{4.3.27}$$

with a natural metric

$$g_U^{Gr(p,n)}(X_1,X_2) = tr(X_1^TX_2) \tag{4.3.28}$$

for the tangent vectors $X_1, X_2 \in T_UGr(p,n)$. The corresponding geodesic curve is derived in [58] to be

$$U(t) = U_1\cos(\Theta t) + (I - U_1U_1^T)U_2F\sin(\Theta t) \tag{4.3.29}$$

where Θ, F, $\cos(\Theta t)$ and $\sin(\Theta t)$ are $p \times p$ diagonal matrices whose i-th diagonal components are θ_i, $\sin\theta_i$, $\cos(\theta_i t)$ and $\sin(\theta_i t)$. Here, θ_i is defined by $\theta_i = \mathrm{acos}\sigma_i$ where σ_i is i-th singular value of $U_1^T U_2$.

A popular Riemannian metric on $P(p)$ at $\Sigma \in P(p)$ is given in [59] by

$$g_{\Sigma}^{P(p)}(Y_1, Y_2) = \frac{1}{2}\mathrm{tr}(\Sigma^{-1} Y_1 \Sigma^{-1} Y_2) \tag{4.3.30}$$

where $Y_1, Y_2 \in Sym(p)$, the tangent space of $P(p)$. The square of Σ along the geodesic curve connecting $\Sigma_1 \in P(p)$ and $\Sigma_2 \in P(p)$ is given by

$$\Sigma^2(t) = \Sigma_1 \exp(t \log(\Sigma_1^{-1}\Sigma_2^2\Sigma_1^{-1}))\Sigma_1. \tag{4.3.31}$$

A natural way to define a distance metric of a product space is to superpose the distance metrics of each space. The resultant metric on $S^+(p,n)$ is then defined by

$$g_{(U,\Sigma)}^{S^+(p,n)}((X_1,Y_1),(X_2,Y_2)) = \mathrm{tr}(X_1^T X_2) + k\mathrm{tr}(\Sigma^{-1}Y_1\Sigma^{-2}Y_2\Sigma^{-1}) \tag{4.3.32}$$

where k is a positive scalar and (X_1, Y_1) and (X_1, Y_1) are tangent vectors on $S^+(p,n)$ at (U, Σ). The corresponding squared geodesic length between (U_1, Σ_1) and (U_2, Σ_2) is given by

$$l^2 = \|\Theta\|_F^2 + k\|\log(\Sigma_1^{-1}\Sigma_2^2\Sigma_1^{-1})\|_F^2. \tag{4.3.33}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and Θ is a $p \times p$ diagonal matrix whose i'th diagonal component is $\cos^{-1}\sigma_i$, where σ_i is i'th singular value of $U_1^T U_2$. Note that $0 \leq \cos^{-1}\sigma_i \leq \frac{\pi}{2}$.

The given metric has invariant properties with respect to important group actions, rotation and scaling. The invariance in scaling is particularly beneficial in

practical applications since it guarantees consistent behaviors of robots under different unit systems. While the length in (4.3.33) is *not a distance* in general because the triangle inequality is not guaranteed to hold, it becomes a distance when $p = 1$, which is proven in Appendix A.5.

4.3.2 Rank-One Manipulability Control

Now we return to manipulability matrix A_{robot}. Eigenvalue decomposition of the manipulability matrix A_{robot} gives,

$$A_{\text{robot}} = Q\Lambda Q^T = \sum_{i=1}^{m} \lambda_i u_i u_i^T, \tag{4.3.34}$$

where $\lambda_i (\geq 0) \in \mathbb{R}$ and $u_i \in \mathbb{R}^m$ are i-th eigenvalue and eigenvector, respectively. One can interpret (4.3.34) that each of the matrices $u_i u_i^T \in \mathbb{R}^{m\times m}$ is a rank-one basis of A_{robot} with the weight λ_i. If any of λ_i is zero, the robot is in kinematic singularity and the matrix become positive semi-definite.

Thus we formulate our manipulability control method using the basis of the manipulability matrix with the biggest eigenvalue λ_1. One advantage of this method is that the value of λ_1 is always non-zero as long as the J^c is non-zero. Let $\tilde{A}_{\text{robot}}$ denote the weighted major basis of A_{robot} as

$$\tilde{A}_{\text{robot}} = \lambda_1 u_1 u_1^T \in S^+(1,m). \tag{4.3.35}$$

By selecting a target direction $u_{\text{tar}} \in \mathbb{R}^m$ and a target magnitude λ_{tar}, one can define a rank-one target manipulability by

$$A_{\text{tar}} = \lambda_{\text{tar}} u_{\text{tar}} u_{\text{tar}}^T. \tag{4.3.36}$$

Note that depending on the choice of u_{tar} and λ_{tar}, A_{tar} can be a constant or a function of q. The metric (4.3.32) at $(u,\lambda) \in S^+(1,m)$ and the distance (4.3.33)

between $\tilde{A}_{\text{robot}}$ and A_{tar} are now simplified to

$$dl^2 = du^T du + k(\frac{d\lambda^2}{\lambda^2})^2 \tag{4.3.37}$$

$$l^2 = \left(\cos^{-1}(u_{\text{tar}}^T u_1)\right)^2 + k\left(\log\frac{\lambda_{\text{tar}}}{\lambda_1}\right)^2 \tag{4.3.38}$$

for a positive scalar k. As mentioned in previous subsection, (4.3.38) is a distance which satisfies the triangle inequality.

To reduce the distance for manipulability control, a straightforward way is to control the robot in the direction of negative gradient of (4.3.38) with respect to joint angle q. The gradient is derived as

$$\begin{aligned}\frac{\partial l^2}{\partial q_k} =& \frac{-2\cos^{-1}(u_{\text{tar}}^T u_1)}{\sqrt{1-(u_{\text{tar}}^T u_1)}}\left(u_1^T\frac{\partial u_{\text{tar}}}{\partial q_k} + u_{\text{tar}}^T\frac{\partial u_1}{\partial q_k}\right) \\ &+ 2k\log\frac{\lambda_{tar}}{\lambda_1}\left(\frac{1}{\lambda_{\text{tar}}}\frac{\partial\lambda_{\text{tar}}}{\partial q_k} - \frac{1}{\lambda_1}\frac{\partial\lambda_1}{\partial q_k}\right).\end{aligned} \tag{4.3.39}$$

The derivatives of eigenvector and eigenvalue are given in Appendix A.3. From (4.1.12), the derivative $\frac{\partial A_{\text{robot}}}{\partial q_k}$ is given as

$$\frac{\partial A_{\text{robot}}}{\partial q_k} = \Phi\left(\frac{\partial J^c}{\partial q_k}WJ^{cT} + J^c\frac{\partial W}{\partial q_k}J^{cT} + J^cW\frac{\partial J^{cT}}{\partial q_k}\right)\Phi. \tag{4.3.40}$$

When $W = I$, we need the derivative of J^c since $\frac{\partial W}{\partial q_k}$. If $W = M^{-1}$, the derivative is given as $\frac{\partial W}{\partial q_k} = -M^{-1}\frac{\partial M}{\partial q_k}M^{-1}$, where the derivative of M is required as well. The analytic derivatives of J^c and M can be efficiently calculated by utilizing Lie-theoretic formulation of robot kinematics and dynamics as detailed in Appendix A.2. These analytic computations are essential for numerical stability, especially in the vicinity of singular configuration.

4.4 Collision Avoidance with Preparatory Action

Real-time collision avoidance is one of the very classical and, at the same time, an active research area for safety. A large portion of the methods belongs to the category that generates artificial repulsive force in Cartesian space along the negative gradient direction of a repulsive potential. As we mentioned earlier, the robot cannot react to the object if the repulsive force is aligned with the direction of the robot's kinematic singularity. In this section, we propose a real-time collision avoidance algorithm that utilizes the manipulability control algorithm of Section 4.3 as an add-on to existing repulsive potential so the robot can configure its posture reactive to the object. Here we only consider the translational part of Jacobian J_v^c for simplicity since it is dominant in safety analysis. The minimum requirement for applying our method is that the Jacobian J_v^c is not a zero matrix, which is not a demanding requirement for a serial manipulator.

4.4.1 Repulsive and Preparatory Potential Functions

In this section we utilize the position-dependent repulsive potential $V_r(x_{rel})$ described in [18], which is

$$V_r(x_{rel}) = \frac{1}{\|x_{\text{rel}}\|}. \tag{4.4.41}$$

Given the relation

$$\dot{x}_{sc} = R_{sc} v^c = R_{sc} J_v^c \dot{q}, \tag{4.4.42}$$

the gradient of a repulsive potential function $V_r(x_{\text{rel}}, v_{\text{rel}})$ with respect to q is derived by applying the chain rule as

$$\frac{\partial V_r}{\partial q_k} = \frac{\partial V_r}{\partial x_{sc}}^T R_{sc} J_{v,k}^c \tag{4.4.43}$$

where $J_{v,k}^c$ is k'th column of J_v^c.

Name	Potential V_p
PSD dist (proposed)	$\left(\cos^{-1}(u_{\text{tar}}^T u_1)\right)^2 + k\left(\log \frac{\lambda_{\text{tar}}}{\lambda_1}\right)^2$
Vector norm	$\|\lambda_{\text{tar}} u_{\text{tar}} - \lambda u\|_2^2$
Frobenius norm	$\|A_{\text{tar}} - \tilde{A}_{\text{robot}}\|_F^2$
Direc. man.	$-\sqrt{u_{\text{tar}}^T A_{\text{robot}} u_{\text{tar}}}$
Squared direc. man.	$-u_{\text{tar}}^T A_{\text{robot}} u_{\text{tar}}$
Isotropic man.	$-\sqrt{det(A_{robot})}$

Table 4.1: List of preparatory potentials.

Besides the repulsive potential, the robot needs to configure itself to increase its moving capability along u_{tar}, represented by directional manipulability μ as

$$\mu = \sqrt{u_{\text{tar}}^T A_{\text{robot}} u_{\text{tar}}}. \tag{4.4.44}$$

The corresponding joint motion is called *preparatory action*, generated by the preparatory potential function V_p. If we have λ_{tar} and u_{tar}, we can use the distance on $S^+(1,3)$ of (4.3.38) as V_p. Other types of preparatory potential can be chosen as well; a list of candidates is on Table 4.1. Amongst them, the squared directional manipulability coincides with the inverse of reflected mass. Performances of these types of preparatory actions will be compared in our experiments. Note that here we choose the value λ_{tar} to be always greater than the greatest achievable directional manipulability $\lambda_{\max} = \sup_q \lambda_1$ to make sure that the robot is controlled in a way to increase its directional manipulability for all time.

Since we define a measure of danger as the repulsive potential, the direction u_{tar} can be defined as the gradient of V_r with respect to Cartesian position x_{sc}

with unit length. Denoting $\psi = \frac{\partial V_r}{\partial x_{sc}}$, the target direction and its derivative with respect to q_k are given by

$$u_{\text{tar}} = \frac{\psi}{\|\psi\|}, \tag{4.4.45}$$

$$\frac{\partial u_{\text{tar}}}{\partial q_k} = \left(\frac{1}{\|\psi\|} I_{3\times 3} - \frac{1}{\|\psi\|^3} \psi\psi^T \right) \frac{\partial \psi}{\partial q_k} \tag{4.4.46}$$

where

$$\frac{\partial \psi}{\partial q_k} = \frac{\partial^2 V_r}{\partial x_{sc}^2} R_{sc} J^c_{v,k}. \tag{4.4.47}$$

4.4.2 Hierarchical Control and Task Relaxation

In our collision avoidance framework, a certain level of abstraction for the robot's exterior shape is required for efficient computation. We designate n_c control points on the robot's skeleton as shown as red dots in Fig. 1.2, and define repulsive and preparatory potentials for each control point c_i, which we denote V_r^i and V_p^i, respectively.

To effectively blend the avoidance motion with a given nominal task, we utilize a hierarchical control framework with a simple kinematic controller that gives target velocity input $\dot{q}_{\text{input}}$ to the robot. The hierarchical controller consists of two layers. The first layer takes care of the activation of the avoidance action based on the level of danger represented by the repulsive potential V_r. When the avoidance action is active, the second layer decides whether to activate the preparatory action based on the value of directional manipulability μ. The resulting velocity input $\dot{q}_{\text{input}}$ is then computed by blending a nominal task input $\dot{q}_t$, the repulsive input $\dot{q}_r$, and the preparatory input $\dot{q}_p$, where the activation and the blending ratio of each input is computed using V_r and μ.

The activation regions of each input in (V_r, μ) space are visualized in Fig. 4.5 in the case of a single control point. The threshold variables $\mu_{\text{threshold}}$, $V_{r,\text{danger}}$,

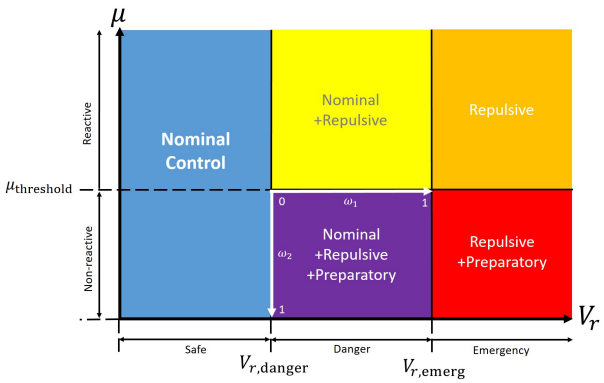


Figure 4.5: Activation regions of repulsive and preparatory actions in (V_p, μ) plane in case of single control point.

and $V_{r,\text{emerg}}$ and relaxation parameters ω_1 and ω_2 will be introduced later again. In the presence of multiple control points the controller makes decision based on the control point in the worst situation.

For a nominal task defined in the joint space, the first hierarchy is expressed as

$$\dot{q}_{\text{input}} = (1 - \omega_1)\dot{q}_t + \omega_1 \dot{q}_a \tag{4.4.48}$$

where $\dot{q}_a$ is an avoidance input, which is yet an unknown mixture of the repulsive and preparatory actions, and $\omega_1 \in [0, 1]$ is a relaxation parameter that is proportional to the level of danger. If the nominal task is defined in Cartesian space, we use a null-space projection of Jacobian with task relaxation [60], given as

$$\dot{q}_{\text{input}} = (1 - \omega_1)\dot{q}_t + \omega_1 (I - J^{\dagger} J)\dot{q}_a. \tag{4.4.49}$$

When the robot is considered safe, the controller doesn't activate the avoidance action by setting $\omega_1 = 0$. The avoidance action is activated by $\omega_1 > 0$ when the robot is considered to be in *danger*. When it's in *emergency*, the controller

deactivates the nominal task and fully activates the avoidance action by setting $\omega_1 = 1$.

The danger or emergency status is assigned on each of the control points and judged by the value of each repulsive potential V_r^i. By defining the thresholds $V_{r,\text{danger}}^i$ and $V_{r,\text{emerg}}^i$ such that $V_{r,\text{emerg}}^i > V_{r,\text{danger}}^i > 0$, the relaxation parameter ω_1 is computed by

$$\omega_1 = \begin{cases} 0 & \text{if } \eta < 0 \\ \eta & \text{if } 0 \leq \eta < 1 \\ 1 & \text{if } \eta \geq 0 \end{cases} \tag{4.4.50}$$

where η is given by

$$\eta = \max_i \frac{V_r^i - V_{r,\text{danger}}^i}{V_{r,\text{emerg}}^i - V_{r,\text{danger}}^i}. \tag{4.4.51}$$

The second layer is for avoidance input $\dot{q}_a$. It is a superposition of the repulsive action and preparatory action defined as

$$\dot{q}_r = -k_r \sum_{i=1}^{n_c} \frac{\partial V_r^i}{\partial q} \tag{4.4.52}$$

$$\dot{q}_p = -k_p \sum_{i=1}^{n_c} \frac{\partial V_p^i}{\partial q} \tag{4.4.53}$$

where $k_r, k_p > 0$ are gain for repulsive and preparatory action, respectively. Note that $\dot{q}_p$ only aims at increasing the directional mainipulability and may drive the robot closer to the object, which increases the risk of collision. Since the actual avoidance motion is generated by $\dot{q}_r$, we project $\dot{q}_p$ onto space that does not increase the level of danger. More precisely, $\dot{q}_p$ is projected onto the null space of $\dot{q}_r$ if $\dot{q}_p^T \dot{q}_r < 0$. The projection matrix P is given by

$$P = \begin{cases} I - \frac{\dot{q}_r^T \dot{q}_r}{\dot{q}_r^T \dot{q}_r} & \text{if } \dot{q}_p^T \dot{q}_r < 0 \\ I & \text{otherwise} \end{cases} \tag{4.4.54}$$

and the avoidance input is defined as

$$\dot{q}_a = (1 - \omega_2)\dot{q}_r + \omega_2 P \dot{q}_p \tag{4.4.55}$$

where $\omega_2 \in [0, 1]$ is the relaxation parameter for second layer. One thing to remark is that the second layer is activated by the value of directional manipulability μ^i since it represents the moving capability to the target direction that is independent of the choice of preparatory potential function. Letting $\mu^i_{\text{threshold}}$ denotes the activation threshold for control point i, we compute ω_2 by

$$\omega_2 = \begin{cases} 0 & \text{if } \zeta < 0 \\ \zeta & \text{otherwise} \end{cases} \tag{4.4.56}$$

where

$$\zeta = \max_i \frac{\mu^i - \mu^i_{\text{threshold}}}{-\mu^i_{\text{threshold}}}. \tag{4.4.57}$$

4.5 Experiments

In this section, the proposed manipulability control algorithm and collision avoidance algorithm are validated in a set of numerical simulations and hardware experiments. For manipulability control, experiments are to control the robot in the negative direction of gradients of the preparatory potentials for either constant u_{tar} or varying u_{tar} of (4.4.46) from the repulsive potential associated with an object. The behavior of the proposed PSD measure is compared to the other preparatory potentials shown in Table 4.1. Note that the derivatives of the other preparatory potentials can also be computed analytically using the equations from Appendix A.2, A.3, and using Jacobi's formula for isotropic manipulability as described in [61].

Similarly, experiments on collision avoidance algorithms are conducted with a moving object. The performance of the proposed PSD distance is compared to the

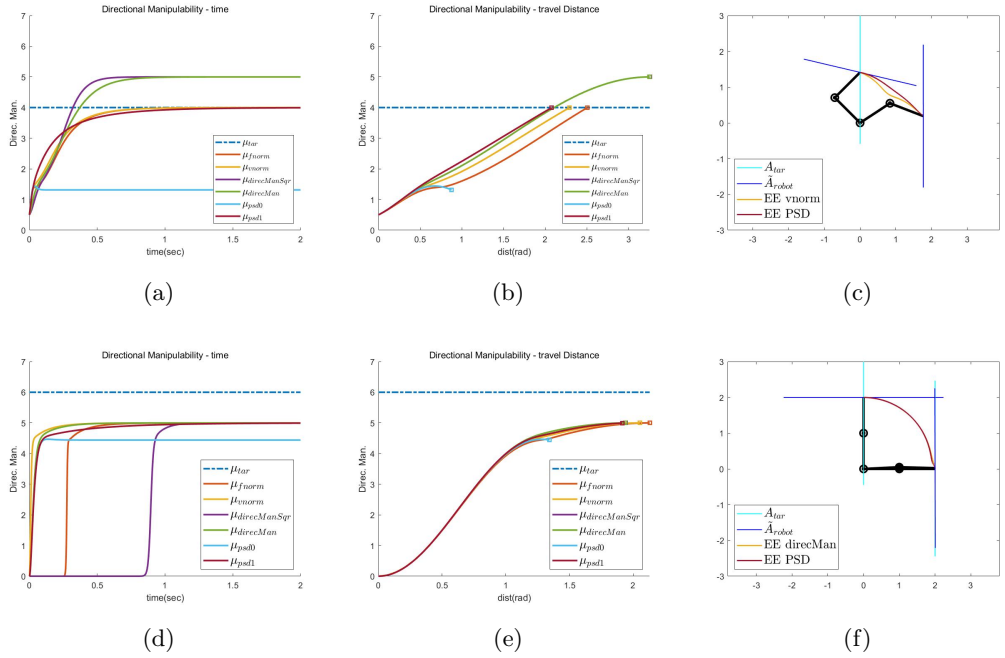
preparatory potentials shown in Table 4.1. The activation of the repulsive and the preparatory action is done by the activation rule illustrated in Fig. 4.5.

The control input is calculated every time step with $1ms$ interval. Simulations are conducted using *MATLAB* for 2-dimensional case and Visual Studio 2013 with C++ for 3-d case in PC with i7-8656U (1.80GHz) and 16GB RAM. For hardware experiments, a PC with Intel Celeron Braswell soc(1.6GHz) and 4GB RAM is used and the EtherCat master program is implemented in Linux with Xenomai real-time environment.

4.5.1 Manipulability Control

The experiment with constant u_{tar} and λ_{tar} is performed using two-dof planar robot of which the length of each link is one. If a control point is attached on the end-effector and we set $W = I$ to consider the kinematic case, the supremum and infimum of λ_1 are 5 and 1, respectively. For proposed PSD distance measure, the parameter k is set to 1. The result of PSD distance with $k = 0$ is also presented to see what happens if we only consider the angle between u_1 and u_{tar}.

Throughout the experiments, the directional manipulability μ of (4.4.44) is chosen as the performance measure, since we aim at maximizing the moving capability along the direction. If a similar level of μ is achieved, elapsed time and traveled distance until the convergence of μ are used to compare the results. So the two types of graph is provided, μ versus time and distance, to exactly see how the robot behaves with the manipulability control input. Note that the gain k_p for each preparatory potential is tuned to have similar time-convergence behavior for a non-singular case.



(a) (b) (c)

(d) (e) (f)

Figure 4.6: Performance graph of manipulability tracking of two-dof manipulator. (a)∼(c) are graphs for non-singular case whose initial pose is $\left[\frac{3\pi}{4}, -\frac{\pi}{2}\right]$, target value is $\lambda_{\text{tar}} = 4$, and $u_{\text{tar}} = (0, 1)^T$. (d)∼(f) are graphs for singular case whose initial pose is $\left[\frac{\pi}{2}, 0\right]$, target value is $\lambda_{\text{tar}} = 6$, and $u_{\text{tar}} = (0, 1)^T$.

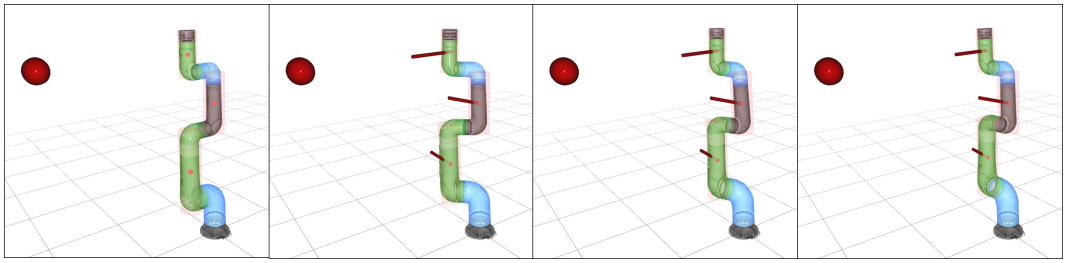
The result is shown in Fig. 4.6 for two distinctive cases, one is from non-singular to non-singular configuration and the other is from singular to singular configuration. For the experiment with non-singular initial/final configurations, which we set $\lambda_{\text{tar}} = 4$ and $u_{\text{tar}} = (0, 1)^T$, three measures showed successful result in μ versus time graph, which are proposed PSD distance, vector norm, and Frobenius norm. The directional manipulability and its squared measure also increase the value of μ rapidly, but μ goes above λ_{tar} since the measures are not dependent

on λ_{tar}. PSD distance with $k = 0$ converges with a very short time and traveled distance, but it converges to the wrong value since this is also not dependent on λ_{tar}.

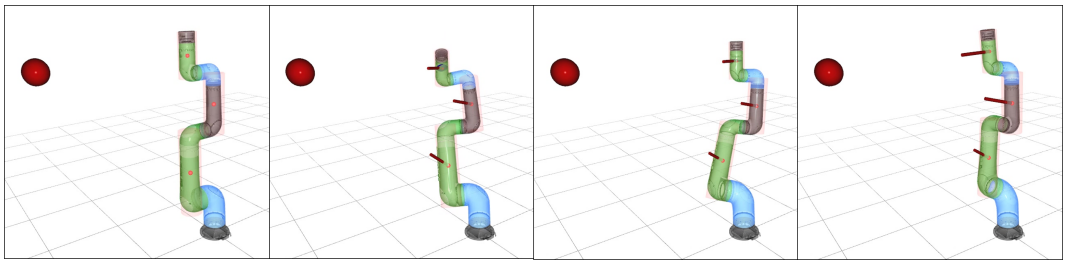
In the μ versus distance graph in Fig. 4.6b, the proposed PSD distance has shorter distance traveled until convergence than the other two norm measures. The phenomenon can also be observed from Fig. 4.6c, which shows the end-effector path of PSD distance and vector norm measure. The path with PSD distance tends to track both the magnitude and direction of target manipulability at the same time while the vector norm tends to only increase combined value, resulting in wavy end-effector path.

In the experiment with singular initial/final configurations, which we set $\lambda_{\text{tar}} = 6$ and $u_{\text{tar}} = (0, 1)^T$, proposed PSD distance, vector norm, and directional manipulability measure show similar behavior. The result of PSD distance shows a shorter distance traveled, but the difference with the other two is marginal. In contrast, Frobenius norm and squared directional manipulability measure have finite time delay until they are activated. This is because they have zero gradient when the target direction coincides with the singular direction which is mentioned in Section. 4.2.

For the manipulability control with a variable u_{tar} associated with an object, a six-dof cooperative manipulator, Indy7 [62], is used with three control points assigned on second, third, and fifth links as illustrated in Fig. 1.2. Here we choose $W = M^{-1}$ to control the robot in a way to decrease the reflected mass for safe manipulation, as described in [32]. The suprema and infima of biggest eigenvalues of control points are found by exhaustive search and λ^i_{tar} are set to be higher than the suprema to increase μ^i as much as possible, similar to the singular case of two-dof robot. The parameter k for proposed PSD measure is also set to 1. We



(a)

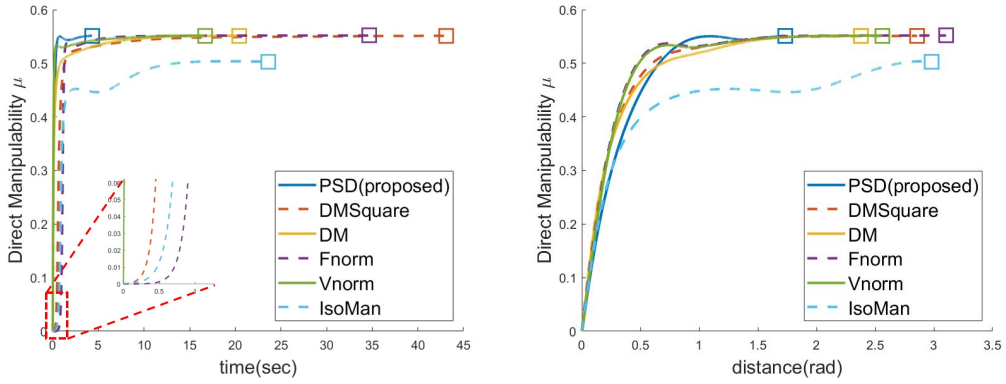


(b)

Figure 4.7: Snapshots of preparatory motion with respect to an object. (a) PSD distance, (b) vector norm.

set the object static at where the relative position vector coincides with a singular direction of the robot. Although the control inputs for all three control points are calculated and applied, we only see the behavior of the control point on the fifth link for clarity.

The result for a static object is shown in Fig. 4.8. Compare to the other preparatory potentials, the proposed PSD measure shows significantly shorter time and traveled distance until convergence and swiftly escaped from a singular configuration with non-zero gradient. All the other measures continued moving even they reached a certain level of μ. Fig. 4.7 shows the snapshots of the tracking motions



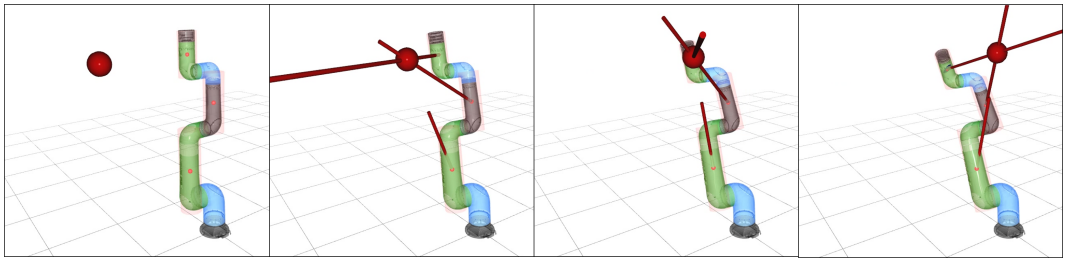
(a) (b)

Figure 4.8: Performance graphs of manipulability control of a six-dof manipulator at the control point on link 5. u_{tar} is associated with the repulsive potential of an object. (a) μ versus time graph, (b) μ versus distance graph.

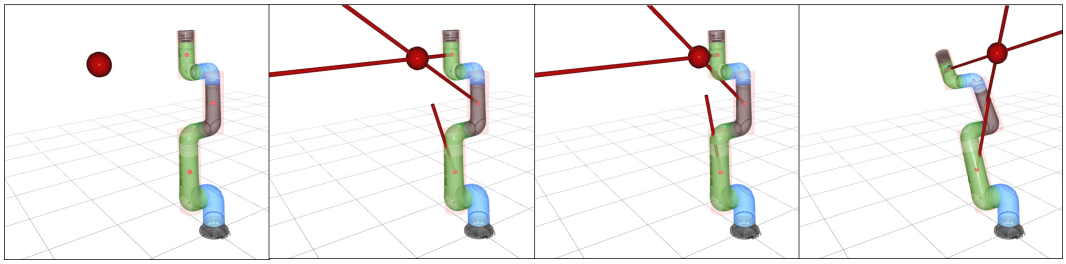
with PSD distance and vector norm, where we can observe that PSD distance generates efficient motion while vector norm generates the motion that sway back. In case of squared directional manipulability measure, Frobenius norm measure, and isotropic manipulability measure increase the μ value very slowly at the beginning stage as shown in the zoomed-in figure in Fig. 4.8a, which agrees with the result from constant u_{tar} case of two-dof robot. The isotropic manipulability measure is the only one that doesn't get close to the maximum value of μ, which shows the importance of direction dependency.

4.5.2 Collision Avoidance

The experiment on collision avoidance uses the same environment with the case of manipulability control with a static object, except now the object is approaching



(a)



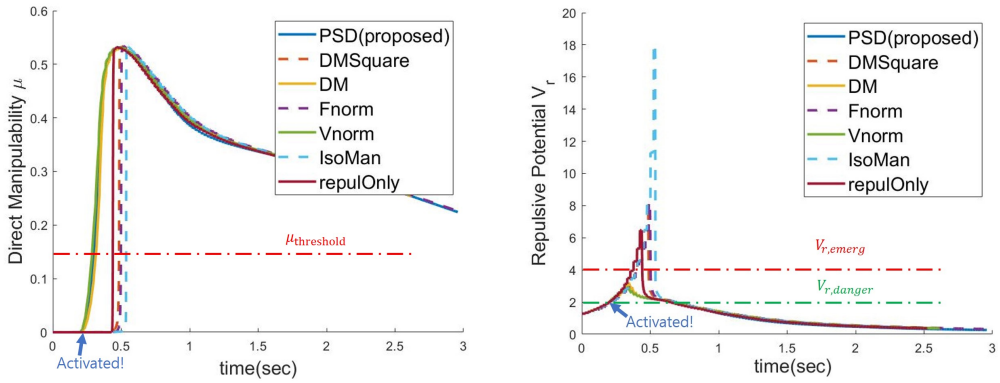
(b)

Figure 4.9: Collision avoidance motion of Indy7. (a) motion with repulsive + preparatory avoidance, (b) motion with repulsive-only avoidance.

to the robot from the singular direction. The nominal task is to move towards the initial configuration of the robot in joint space, which means we use (4.4.48) for task relaxation. We compare the result of different preparatory potentials with the same activation rule and the motion of repulsive-only avoidance algorithm is given as a reference. The experiments are performed with two different robots; Indy7 and the ModMan introduced in Chapter. 2.

4.5.2.1 Indy7

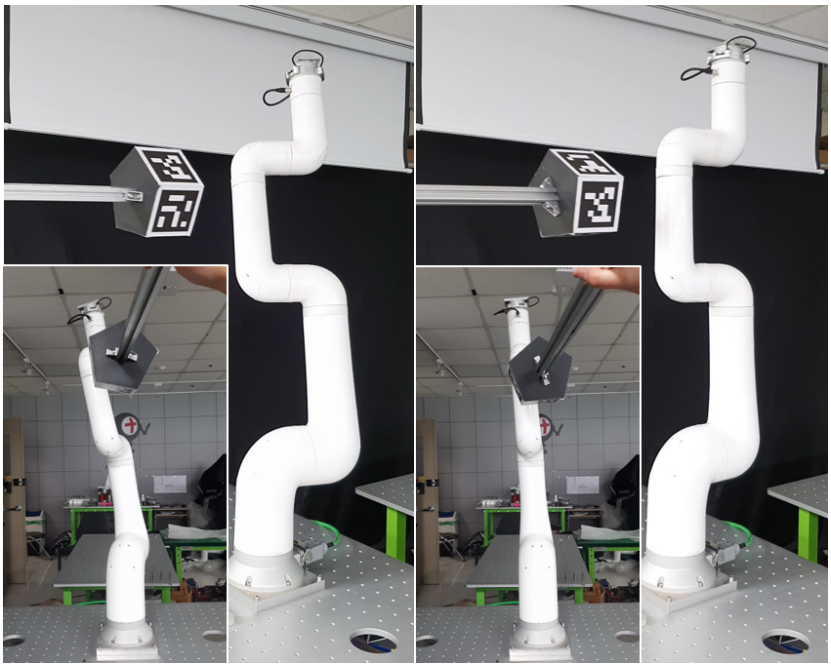
The result for the control point on the fifth link of Indy7 is shown in Fig. 4.10. Note that we provide μ versus time and V_r versus time graph because the value



(a) (b)

Figure 4.10: Performance graphs of collision avoidance of a six-dof manipulator at the control point on link 5. An object approached to the robot from singular direction. (a) μ versus time graph, (b) V_r versus time graph.

of repulsive potential is directly related to safety and time is much more important than traveled distance. As the object gets closer to the robot, V_r increases. As soon as V_r reaches $V_{\mathrm{r,danger}}$, avoidance action $\dot{q}_a$ starts to be blended with nominal task. Since μ is zero, ω_2 is one which means the avoidance motion fully consists of the preparatory action. In cases of the proposed PSD measure, directional manipulability, and vector norm measure, the preparatory actions start to increase the μ value immediately after the activation and successfully avoid collision, not reaching $V_{\mathrm{r,emerg}}$(three graphs almost overlapped). In contrast, the other preparatory measures fail to increase their μ before collision since these have zero gradient at the singular case. They give even worse result than repulsive-only case, since ω_2 remains close to one and repulsive motion cannot be activated properly. Fig. 4.9a

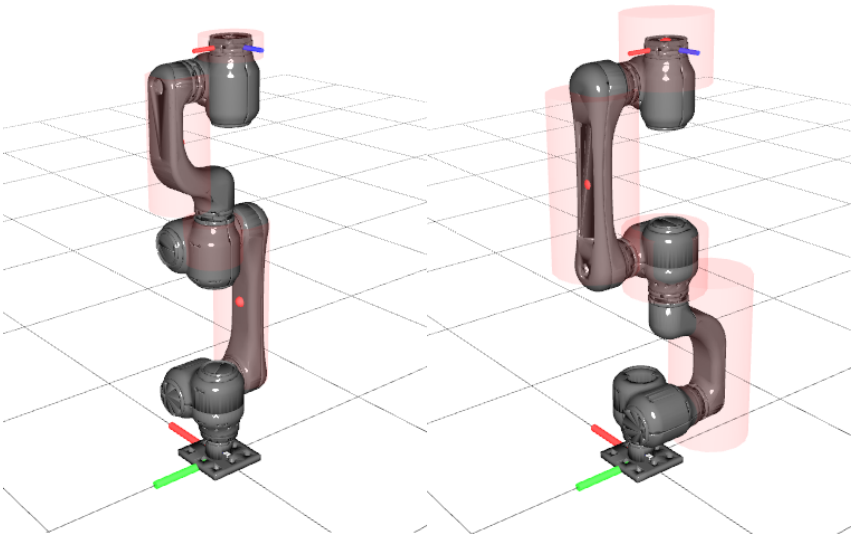


(a) (b)

Figure 4.11: Snapshot of avoidance motion of the real hardware, which is in *danger* state. (a) repulsive + preparatory motion, (b) repulsive-only motion.

and Fig. 4.9b show the avoidance motion of the robot with the PSD distance measure and the isotropic manipulability measure, respectively. The motion with PSD distance immediately drives robots to be reactive to the repulsive potential while the motion with isotropic manipulability measure doesn't generate any motion until collision occurs.

Experiments on collision avoidance are performed with real Indy7 hardware as shown in Fig 4.11, where an object is manually approached to the robot from a singular direction. The robot under repulsive-only avoidance motion is initially not reactive to the object since the gradient of the repulsive potential is very small for



(a) (b)

Figure 4.12: Two different topologies of ModMan for collision avoidance experiments. (a) regular topology, (b) topology that link 2 and 5 are swapped.

an object placed in a singular direction. As the object approaches, the repulsive-potential rapidly increases and the robot suddenly moves to decrease the potential. On the other hand, The avoidance motion augmented with the preparatory motion immediately changes the robot's posture reactive to the object, which leads to a smoother avoidance motion.

4.5.2.2 ModMan

The same experiments are performed with the ModMan, configured with two different topologies as shown in Fig 4.12, with four control points. In the case of the ModMan, the object directions rarely be exactly singular directions for all control

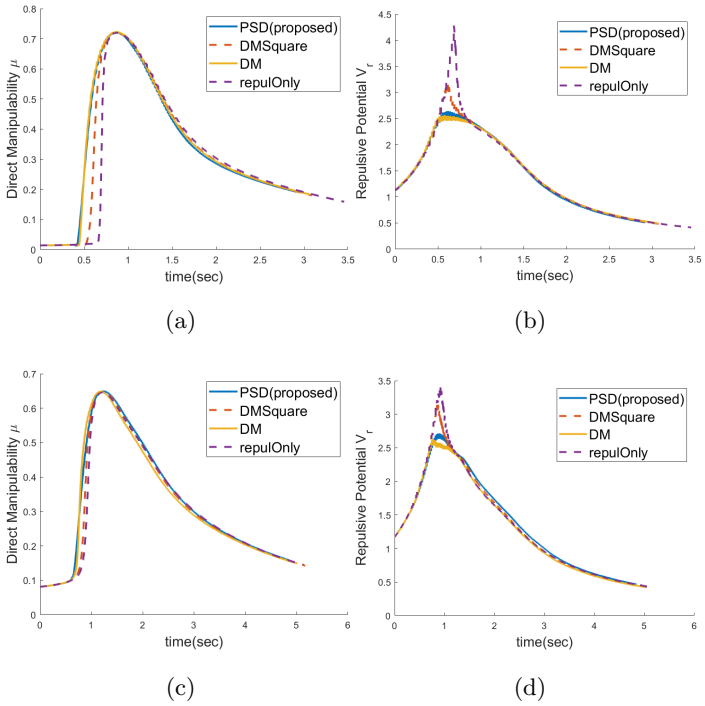


Figure 4.13: Performance graphs from collision avoidance of two six-dof manipulator topologies of ModMan at the control point on link 5. An object approached to the robot from near-singular direction. (a) and (b) are μ versus time graph and V_r versus time graph of regular topology, (c) and (d) are those of second topology.

points at the same time since the two-dof modules have offset between the joint axes. However, when an active control point has a dominant value of repulsive potential and the approaching direction is near singular in that particular control point, the robot cannot properly react to the object. For both topologies, similar phenomena occur at the control points on link 5 when the objects approach to the robot from $-y$ direction.

The result are shown in Fig. 4.13, which shows μ and V_r with respect to time.

Table 4.2: Time consumed to calculate one step

	n_c	1	5	10	50
Time(ms)	Repu. Only	0.13	0.31	0.45	1.37
	Repul. + Kin	0.17	0.36	0.54	1.82
	Repul. + Dyn	0.256	0.47	0.64	2.72

Although the graphs are not as dramatic as those of Indy7 because of the offsets at the two-dof modules, tendencies are similar. Motions for avoidance with the squared directional manipulability and the repulsive-only avoidance show slow response to the object, result in a high value of V_r. In contrast, repulsive + preparatory with the PSD measure and the directional manipulability start to avoid the object as soon as V_r exceeds the threshold.

4.5.2.3 Computation Time

To show that our proposed algorithm is computationally efficient enough for real-time application, we measure the time consumed to compute each step of the collision avoidance algorithm with respect to the number of control points. To consider worst-case performance, all the control points are located at the tool link where the points have the most computations.

Table 4.2 shows the average computation times of repulsive-only case, repulsive plus preparatory with kinematic PSD measure, and repulsive plus preparatory with dynamic PSD measure. Compare to repulsive-only motion, adding preparatory action demands quite a computation power. However, the computation using

less than 10 points guarantees a 1 $msec$ update period, which is enough for real-time collision avoidance.

4.6 Conclusion

Beginning with the premise that a preparatory action plays an important role in collision avoidance, particularly near singular directions, in this thesis we propose a potential function to achieve the twin objectives of minimizing impulse of the collision while maintaining sufficient manipulability in the needed directions for collision avoidance. Our main contribution in this chapter lies in defining a distance between current and target manipulability ellipsoid with the distance on $S^+(1, m)$ by only considering their major axes. The PSD distance measure has many desirable properties: it is invariant under rotation and scaling, it is well-defined near the kinematic singularity, and it is applicability to low-dof robots. Using the PSD distance as a potential function in joint space enables us to generate the control input to track target manipulability. A non-zero gradient in the vicinity of the kinematic singularity of the robot guarantees a fast escape from its singular position. Analytic formulations of all the gradients offer numerical stability and enable fast computations for real-time applications. The preparatory action from PSD distance is used as an add-on to a hierarchical collision avoidance algorithm and compared with other potential-based collision avoidance algorithms qualitatively and quantitatively in numerical simulation experiments. The algorithm is also verified with real hardware including actual measurements of the computation time in the master controller of the robot.

The proposed framework can be generalized to avoid self-collisions and collisions with multiple objects. Furthermore, the preparatory actions using the PSD

distance is applicable to many other safety-related applications including velocity shaping and collision-free path planning.

5

Robust Collision Avoidance Using Invariance Control with Velocity-Dependent Constraints

Relative speed is important for robot safety. As shown in [32], the amount of impact transmitted to an object is proportional to the speed as well as the weight of the object. In Chapter 4, we used a repulsive potential that is dependent only on the relative position. As in [44, 45], there are repulsive potentials that are dependent on both relative position and relative velocity. Now that each constraint is dependent on $\dot{q}$ as well as q, a new control method is needed instead of kinematic control.

A suitable method is to obtain the torque input from velocity-dependent constraints using a nonlinear control algorithm. There are studies that obtain control inputs that satisfy a set of inequality constraints using forward invariance of

the admissible set [43, 63]. While the safety cannot be guaranteed with the hierarchical control introduced in the previous chapter, these methods can be used to mathematically guarantee the safety of the robot. The invariance control described in [43] is suitable for our application since the method can take into account multiple constraints seamlessly.

An important condition is necessary when using invariance control; the initial state of the robot should not seriously violate constraints. However, in the case of the preparatory potential, there is a possibility that it is far from the threshold from the beginning depending on the state of the robot. In the worst case of Fig 1.2, the directional manipulability value for the approaching direction of the robot is zero, which is quite far from the threshold. Therefore, there is a need for a way to escape the robot from this situation and quickly send the robot state in the admissible set.

In this chapter, we propose a nonlinear control method for robust collision avoidance using velocity-dependent repulsive constraints. Invariant control is used to mathematically guarantee the safety of the robot. We propose a non-linear control method that generates an input that can escape from the state that the robot violates the preparatory constraint significantly.

The chapter is organized as follows. In Section 5.1, we briefly review the input-output linearization of a nonlinear system with constraints. Section 5.2 describes the invariance framework. In Section 5.3, we describe velocity-dependent repulsive and preparatory safety constraints and a method to find corrective control when the robot violates constraint significantly. A set of experiments for safe and dangerous initial states is demonstrated in 5.4. The conclusion is presented in Section 5.5.

5.1 Input-Output Linearization

In this section, we review the input-output linearization of a nonlinear robotic system introduced in [64]. For a state vector $X \in \mathbb{R}^N$, a nonlinear system equation is of the form

$$\dot{X} = f(X) + g(X)U. \tag{5.1.1}$$

Since safety criteria are given as inequality constraints, we define l virtual output functions as

$$y_i = h_i(X) \leq 0, y_i \in \mathbb{R}, 1 \leq i \leq l. \tag{5.1.2}$$

or in vector form

$$y = h(X) \preceq 0, y \in \mathbb{R}^l. \tag{5.1.3}$$

f, g and h_i are assumed to be piece-wise smooth to apply input-output linearization.

The rigid body dynamics of n-dof manipulator,

$$M(q)\ddot{q} + c(q, \dot{q}) + g(q) = \tau \tag{5.1.4}$$

where $M(q) \in \mathbb{R}^{n \times n}$ is mass matrix of link inertia in configuration space, $c(q, \dot{q}) \in \mathbb{R}^n$ is Coriolis term, $g(q) \in \mathbb{R}^n$ is gravity term, and $\tau \in \mathbb{R}^n$ is load torque. With $X = \left(q^T, \dot{q}^T\right)^T$ and $U = \tau$, equation 5.1.4 can be represented as 5.1.1 with

$$f(X) = \begin{pmatrix} \dot{q} \\ -M^{-1}(q)\left(c(q, \dot{q}) + g(q)\right) \end{pmatrix} \tag{5.1.5}$$

$$g(X) = \begin{pmatrix} 0 \\ -M^{-1}(q) \end{pmatrix}. \tag{5.1.6}$$

If we choose $U = \ddot{q}$, the system is represented as a linear system with

$$f(X) = \begin{pmatrix} \dot{q} \\ 0 \end{pmatrix} \tag{5.1.7}$$

$$g(X) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{5.1.8}$$

Depending on the applications, the robots are constrained by many kinds of conditions. In [65], a position-dependent constraint is used in invariance control framework to avoid collision with an object. However, relative velocity between the robot and the object is important in safe manipulation because it is directly related to the magnitude of impulse as introduced in Chapter 4. If velocity-dependent output functions which are dependent on q and $\dot{q}$ are considered, relative degrees of all constraints become one. When $U = \tau$, the full-state output linearization is given as

$$\begin{aligned} \dot{y} &= \frac{\partial h}{\partial \dot{q}}\ddot{q} + \frac{\partial h}{\partial q}\dot{q} \\ &= \frac{\partial h}{\partial \dot{q}}M^{-1}\tau - \frac{\partial h}{\partial \dot{q}}M^{-1}\left(c(q,\dot{q}) + g(q)\right) + \frac{\partial h}{\partial q}\dot{q} \\ &= A(q,\dot{q})\tau - b(q,\dot{q}). \end{aligned} \tag{5.1.9}$$

When $U = \ddot{q}$, the full-state output linearization is given as

$$\begin{aligned} \dot{y} &= \frac{\partial h}{\partial \dot{q}}\ddot{q} + \frac{\partial h}{\partial q}\dot{q} \\ &= A(q,\dot{q})\ddot{q} - b(q,\dot{q}). \end{aligned} \tag{5.1.10}$$

Assuming that the robot is position-controlled, we choose $\ddot{q}$ as input for the system in this chapter.

5.2 Invariance Control

As described in [43], invariance control is a mean of controlling systems subject to state inequality constraints. The main idea behind invariance control is to render a subset of the state space forward invariant by mean of a switching controller. Under normal operating conditions, the system is controlled by a nominal controller that achieves the control objectives. When the system trajectory approaches the boundary of the admissible region of the state space, the control is switched to a corrective controller, which drives the system back into the interior of the admissible set. Control is switched back to the nominal controller as soon as the nominal control no longer results in any violation of state space constraints. For the case of relative degree one, the invariant set is defined as

$$S = \{X | y_i(X) \leq 0, i = 1, \cdots, l\} \tag{5.2.11}$$

with boundaries $\partial S = \{X | y_i(X) = 0, i = 1, \cdots, l\}$. The set S is invariant if and only if the control input u satisfies

$$U = \begin{cases} U_{nom} & \text{if } \forall_i \quad y_i(X) < 0 \text{ or } y_i(X) \geq 0 \wedge \dot{y}_i(X, U_{nom}) < 0 \\ U_{corr} & \text{if } \exists_i \quad y_i(X) \geq 0 \wedge \dot{y}_i(X, U_{nom}) \geq 0. \end{cases} \tag{5.2.12}$$

We generate the corrective control input U_{corr} by using the algorithm introduced in in [66]. For the corrective control, only the active constraints are considered. A set of indices of active constraints $\mathcal{I}$ is given as

$$\mathcal{I} = \{i | y_i \geq 0 \wedge \dot{y}_i(X, U_{nom}) \geq 0\}. \tag{5.2.13}$$

Then (5.1.10) for active constraints can be written as

$$y = A_{\mathcal{I}}(q, \dot{q})u - b_{\mathcal{I}}(q, \dot{q}). \tag{5.2.14}$$

The correction control input U_{corr} is generated by solving a quadratic optimization problem of the form

$$\begin{aligned} &\arg\min_{U_{corr}} \|U_{corr} - U_{nom}\|_2^2 \\ &\text{s.t. } A_{\mathcal{I}}(X)U_{corr} - b_{\mathcal{I}}(X) \prec 0, \end{aligned} \tag{5.2.15}$$

which means that the generated U_{corr} satisfies the forward invariance and at the same time becomes the value closest to U_{nom}. If the number of active constraints is less than the degree of freedom, the above optimization problem can be solved analytically as

$$U_{corr} = A_{\mathcal{I}}^{T}\left(A_{\mathcal{I}}A_{\mathcal{I}}^{T}\right)^{-1}\left(b_{\mathcal{I}} - A_{\mathcal{I}}U_{nom}\right) + U_{nom}. \tag{5.2.16}$$

5.3 Velocity-Dependent Constraints for Robot Safety

The invariance control of the previous section requires two important conditions: (1) the robot should be in the invariant set at the initial state and (2) $A_{\mathcal{I}}(X) \neq 0$. However, there are cases in which the conditions are violated depending on the robot's posture. In the case of Fig 1.2, $A_{\mathcal{I}} = 0$ for the repulsive potentials. To prevent $A_{\mathcal{I}}$ from becoming zero, one can use preparatory potentials, introduced in Chapter 4, to send the robot to a good posture. However, the values of the preparatory potentials are often outside of the invariant set. In this section, we describe how to generate the corrective control input in this case.

5.3.1 Velocity-Dependent Repulsive Constraints

Many algorithms using the potential field approach only consider the relative Cartesian position $x = x_{sc} - x_{\text{obj}} \in \mathbb{R}^3$. However, the relative Cartesian velocity $\dot{x} = \dot{x}_{sc} - \dot{x}_{\text{obj}}$ should also be considered to properly measure the level of danger. The

Name	Potential V_r
inverse position	$\frac{k}{\|x\|}$
danger field [44]	$\frac{k_1}{\|x\|} + \frac{k_2\left(\gamma\|x\|\|\dot{x}\| - x^T\dot{x}\right)}{\|x\|^3}$
safety field [45]	$kx^T x\left(\gamma - x^T\dot{x}\right)$

Table 5.1: List of repulsive potentials. The parameters k_v and γ are positive scalars.

danger field and safety field proposed in [44] and [45], respectively, are dependent on both the relative position and the relative velocity, and applied to collision avoidance. Examples of a position-only potential function and velocity-dependent repulsive functions are listed in Table 5.1. The detailed conditions for a scalar function to be a danger field are given in Appendix A.6.

Since repulsive functions $h_r(x, \dot{x})$ are defined in Cartesian space, the output linearization is done using (4.4.42) as

$$\begin{aligned}
\dot{h_r} =& \frac{\partial h_r}{\partial \dot{x}}\ddot{x} + \frac{\partial h_r}{\partial x}\dot{x} && (5.3.17)\\
=& \frac{\partial h_r}{\partial \dot{x}} R_{sc} J_v^c \ddot{q} + \frac{\partial h_r}{\partial \dot{x}}\left((\dot{R}_{sc} J_v^c + R_{sc}\dot{J}_v^c)\dot{q} - \ddot{x}_{\text{obj}}\right) \\
& + \frac{\partial h_r}{\partial x}\left(R_{sc} J_v^c \dot{q} - \dot{x}_{\text{obj}}\right) && (5.3.18)\\
=& A_r(q, \dot{q})\ddot{q} - b_r(q, \dot{q}). && (5.3.19)
\end{aligned}$$

The formulations for $\dot{R}_{sc}$ and $\dot{J}_v^c$ can be found in Appendix A.2. For a collision avoidance algorithm with only h_r, the corrective input $\ddot{q}_{\text{corr}}$ can be computed using 5.2.16. However, A_r becomes zero when the object approaches to the robot from singular direction as depicted in Figure 1.2. So we need to change the robot's posture to be reactive to the object using preparatory action described in Chapter 4

5.3.2 Preparatory Constraints

The robot needs to configure itself to increase its moving capability along a direction of interest u_{tar}. Here, we choose $u_{\text{tar}} = \frac{\psi}{\|\psi\|}$ where $\psi = \frac{\partial h_r}{\partial x}$ as in Chapter 4. Since u_{tar} is dependent on $\dot{q}$, a preparatory constraint function h_p is also dependent on $\dot{q}$. Similar to (4.4.46), the time derivative $\dot{u}_{\text{tar}}$ is given as

$$\dot{u}_{\text{tar}} = \left(\frac{1}{\|\psi\|} I_{3\times 3} - \frac{1}{\|\psi\|^3}\psi\psi^T\right)\left(\frac{\partial \psi}{\partial x}\dot{x} + \frac{\partial \psi}{\partial \dot{x}}\ddot{x}\right). \tag{5.3.20}$$

where

$$\frac{\partial \psi}{\partial x} = \frac{\partial^2 h_r}{\partial x^2} \tag{5.3.21}$$

$$\frac{\partial \psi}{\partial \dot{x}} = \frac{\partial^2 h_r}{\partial x \partial \dot{x}}. \tag{5.3.22}$$

Since $h_p(q, \dot{q})$ is usually a mixture of terms defined in joint space and Cartesian space, the output linearization of h_p can have different formulation depending on choice of h_p. For the PSD distance measure, the output linearization is given as

$$h_p = \left(\cos^{-1}(u_{\text{tar}}^T u_1)\right)^2 + k\left(\log\frac{\lambda_{\text{tar}}}{\lambda_1}\right)^2 \tag{5.3.23}$$

$$\dot{h}_p = \frac{-2\cos^{-1}(u_{\text{tar}}^T u_1)}{\sqrt{1-(u_{\text{tar}}^T u_1)^2}}\left(u_1^T \dot{u}_{\text{tar}} + u_{\text{tar}}^T \dot{u}_1\right) - \frac{2k}{\lambda_1}\log\frac{\lambda_{tar}}{\lambda_1}\dot{\lambda}_1. \tag{5.3.24}$$

For simplicity, we denote

$$D = \frac{-2\cos^{-1}(u_{\text{tar}}^T u_1)}{\sqrt{1-(u_{\text{tar}}^T u_1)^2}} \in \mathbb{R} \tag{5.3.25}$$

$$P = \left(\frac{1}{\|\psi\|} I_{3\times 3} - \frac{1}{\|\psi\|^3}\psi\psi^T\right) \in \mathbb{R}^{3\times 3}. \tag{5.3.26}$$

Then (5.3.24) becomes

$$\begin{aligned}\dot{h}_p =& D u_1^T P \frac{\partial \psi}{\partial \dot{x}} R_{sc} J_v^c \ddot{q} - \frac{2k}{\lambda_1}\log\frac{\lambda_{tar}}{\lambda_1}\dot{\lambda}_1 \\ &+ D\left(u_1^T P\left(\frac{\partial \psi}{\partial \dot{x}}(\dot{R}_{sc}J_v^c + R_{sc}\dot{J}_v^c)\dot{q} + \frac{\partial \psi}{\partial x}(R_{sc}J_v^c\dot{q} - \dot{x}_{\text{obj}})\right) + u_{tar}^T \dot{u}_1\right).\end{aligned} \tag{5.3.27}$$

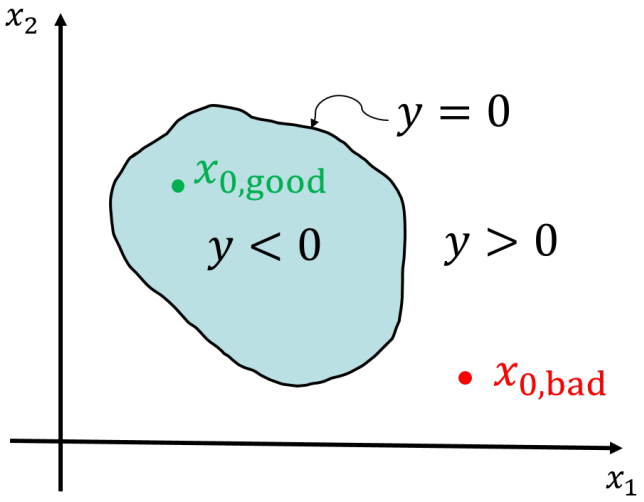


Figure 5.1: Illustration of safe and dangerous initial state x_0.

The formulations of $\dot{R}_{sc}, \dot{J}_v^c, \dot{u}_1$, and $\dot{\lambda}_1$ can be found in Appendix A.2 and A.3. If we represent (5.3.27) in form of $A_p\ddot{q} - b_p$, then

$$A_p = Du_1^T P \frac{\partial^2 h_r}{\partial x_{rel} \partial \dot{x}}. \tag{5.3.28}$$

Remark that A_p becomes zero only when $u_1 = \psi$, which means that the direction of interest coincides with the first principal axis of the manipulability matrix. Since PSD distance have minimum value in this state, the value of h_p is below γ so that the constraint is not active. However, b_p can be zero when the object approaches to the robot in singular direction and $\dot{q} = 0$ at the same time. Since (5.2.16) is not applicable in this state, we present a new control method in next section.

5.3.3 Corrective Control for Dangerous Initial State

The key condition for invariance control is that the robot should be in the invariant set at the initial state. As illustrated in Fig 5.1, if the initial state of the

robot is $x_{0,\text{good}}$, the invariance control generates a control input so that the robot stays within the invariant set. However, if the initial state is $x_{0,\text{bad}}$, the invariance control generates a very large input that the robot cannot follow or an input of magnitude zero, as in the case of preparatory constraints introduced in the previous section. In this section, we propose a method to generate the corrective control for dangerous initial state. The main objective is to decrease h_p as fast as possible. Therefore, the objective can be achieved by generating a control input with the smallest value of $\dot{y}_{prep}$.

If there is no other constraints, the norm of U_{corr} diverges to infinity, which is not desirable. Since the robot normally has the lower and the upper bound for accelerations of all joints, a set of linear inequality constraints can be considered in the optimization. However, according to [67], when there are only linear inequality constraints in linear programming, the solution for the optimization problem activates all constraints with high probability which results in bang-bang control. Here we choose to limit the norm of U, which results in a quadratic constraint.

With above considerations, the corrective control U_{corr} for dangerous initial state is generate by solving following optimization problem,

$$\begin{aligned} &\arg\min_{U_{\text{corr}}} A_p U_{\text{corr}} - b_p \\ &\text{s.t. } U_{\text{corr}}^T U_{corr} < r^2. \end{aligned} \tag{5.3.29}$$

Fig 5.2 shows an example of the optimization problem for a tow-dof case. Since the problem is a simple linear programming, analytic solution for the problem is given as

$$U_{\text{corr}} = \frac{-1}{\sqrt{A_p^T A_p}} A_p. \tag{5.3.30}$$

Note that the activation of the preparatory constraints are check only if the

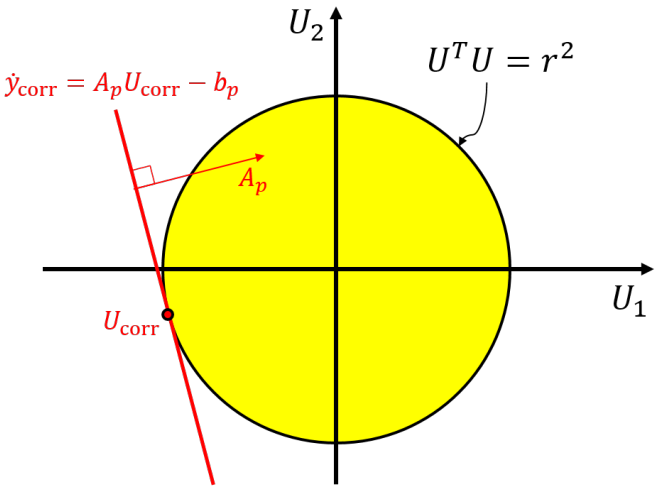


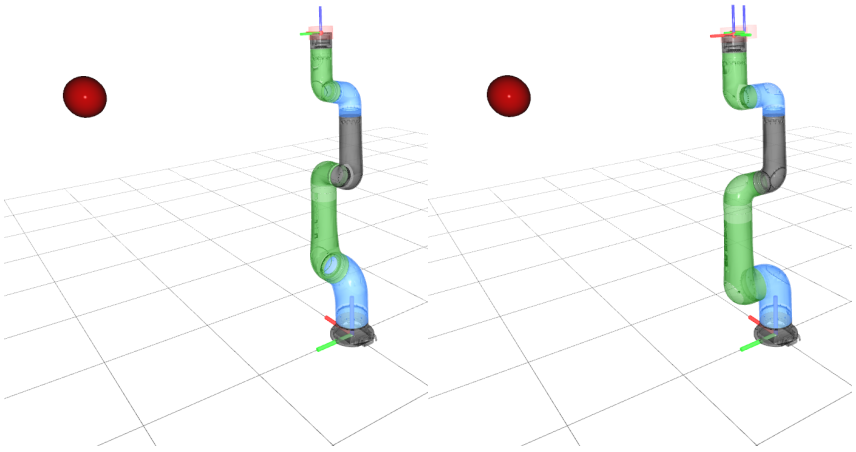
Figure 5.2: Illustration of corrective control for dangerous initial state.

repulsive constraints are active, since the actual avoidance motion is generated by the repulsive constraints.

5.4 Experiment

In this section, the proposed collision avoidance algorithm with invariance control is validated in a set of numerical simulations. In the simulation, the repulsive and preparatory constraints are calculated in a control point attached on the end-effector of a six-dof robot, Indy7. For the repulsive constraint, we choose a danger field of the form in Table 5.1 as

$$h_r(x, \dot{x}) = \frac{k_1}{\|x\|} + \frac{k_2 \left(\gamma \|x\| \|\dot{x}\| - x^T \dot{x}\right)}{\|x\|^3}. \tag{5.4.31}$$



(a) (b)

Figure 5.3: Two different initial states. (a) safe state, (b) dangerous state

Derivatives of h_r with respect to x and $\dot{x}$ are given as

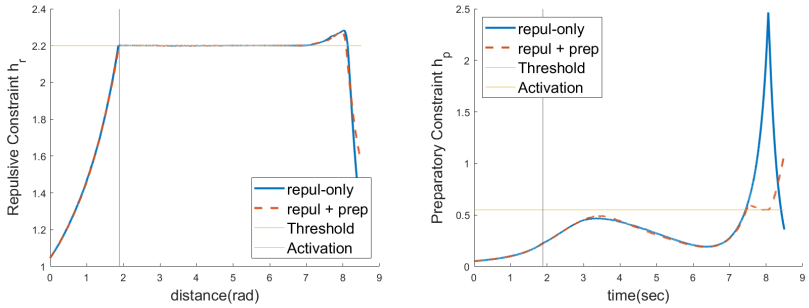
$$\frac{\partial h_r}{\partial x} = -\left(\frac{k_1}{\|x\|^3} + k_2\frac{2\gamma\|\dot{x}\|\|x\| - 3x^T\dot{x}}{\|x\|^5}\right)x - \frac{k_2}{\|x\|^3}\dot{x} \tag{5.4.32}$$

$$\frac{\partial h_r}{\partial \dot{x}} = -\frac{k_2}{\|x\|^3}x + \frac{k_2\gamma}{\|\dot{x}\|\|x\|^2}\dot{x} \tag{5.4.33}$$

$$\begin{aligned}\frac{\partial^2 h_r}{\partial x^2} =& -\left(\frac{k_1}{\|x\|^3} + k_2\frac{2\gamma\|\dot{x}\|\|x\| - 3x^T\dot{x}}{\|x\|^5}\right)I_{3\times 3} \\ &+ k_2\left(\frac{8\gamma\|\dot{x}\|\|x\| - 15x^T\dot{x}}{\|x\|^7}\right)xx^T + \frac{3k_2}{\|x\|^5}\left(x\dot{x}^T + \dot{x}x^T\right)\end{aligned} \tag{5.4.34}$$

$$\frac{\partial^2 h_r}{\partial x \partial \dot{x}} = \frac{k_2}{\|x\|^3}\left(-I_{3\times 3} - \frac{\gamma}{\|x\|\|\dot{x}\|}x\dot{x}^T + \frac{3}{\|x\|^2}xx^T\right). \tag{5.4.35}$$

Simulation is performed for two different initial states: one is inside the invariant set(safe state) and the other is outside the invariant set(dangerous state), as illustrated in Fig 5.3. The object approaches to the robot with $\|x_{\text{obj}}\| = 0.3$ in $-y$ direction. For nominal control, we use a simple PD control that sends the robot to the initial state.

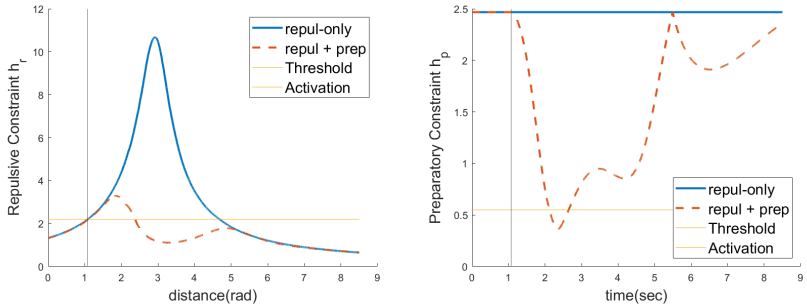


(a) (b)

Figure 5.4: The values of constraints for the safe initial state. (a) repulsive constraint, (b) preparatory constraint

The results for the safe initial posture is illustrated in Fig 5.4. Both the repulsive-only action and the repulsive + preparatory action control the robot so that value of the repulsive constraint does not increase after reaching the threshold. However, in the case of the repulsive-only action, the value of the preparatory constraint rapidly increases at the end of the active state. In contrast, the repulsive + preparatory action ensures that the value of the preparatory constraint does not exceed the threshold at all times in the active state.

The results for the dangerous initial posture is illustrated in Fig 5.5. Since A_r becomes zero for this case, the repulsive-only action does not generate any control input and the object passes through the robot. On the other hand, the repulsive + preparatory action rapidly decreases the preparatory constraint value as soon as it becomes an active state, causing it to fall below the threshold. The value of the preparatory constraint increases quickly again, but this is not a problem since the repulsive constraint is no longer active.



(a) (b)

Figure 5.5: The values of constraints for the dangerous initial state. (a) repulsive constraint (b) preparatory constraint

5.5 Conclusion

In this chapter, a collision avoidance algorithm with invariance control is presented. In order to properly consider the effects of relative speed, a velocity-dependent repulsive function is used as a constraint. Our main contribution is to generate control inputs to avoid collisions even when the object approaches the robot in a dangerous direction. When an object approaches the robot in a dangerous direction, forward invariance cannot be satisfied by invariance control alone, so a preparatory function is added to the constraints so that the robot can move to a reactive posture on the object. Since the preparatory function has a very large value compared to the threshold in the initial state, the value of the preparatory function is rapidly reduced by solving a separate optimization problem.

The performance of the collision avoidance algorithm is validated with a set of simulations. When the robot's initial state is in the invariant set of preparatory

constraints, collisions can be avoided by repulsive-only actions. However, if the initial state is outside the invariant set of the preparatory constraint, the preparatory action must be used together to successfully avoid the collision.

The method introduced in this chapter can be extended to consider multiple control points. In this case, multiple constraints may exist outside the invariant set in the initial state, so a new optimization problem must be defined to generate a control input that sends the state into the invariant set.

6

Conclusion

6.1 Overview of This Thesis

In this thesis, two important features required for the next generation of robots, task adaptability and safety, have been examined. By developing a modular robot system that can be reconfigured, the user can configure a robot suitable for the current task. Regarding robot safety, we have defined new safety measures that are dependent on the robot's current posture and vector in the direction of the object, and control it in a direction that reduces it so that the robot can go to a safer posture. By presenting two different collision avoidance algorithms using this safety measure, it is possible to cope with various operating conditions of the robot. The main contributions of our thesis are summarized as follows.

- **Design of a Reconfigurable Modular Manipulator System**

  Modman, a reconfigurable modular robot system composed of multi-dof bidirectional modules, is developed. The genderless connector is designed to have

sufficient mechanical/electrical capacity required for industrial robots through design optimization. By imposing an offset between the two axes, the developed two-dof modules have a small volume and mass, and thus a high power density. It was confirmed through a repeatability test that the robot assembled in six-dof has comparable performance to the existing fixed architecture robots.

- **A Programming Architecture for Modular Robot Systems**

  A modular robot software architecture that can cope with tree-structured modules has been proposed. We construct a data structure that can represent the properties of each generalized module to suit Lie group kinematics and dynamics. By distinguishing the parameters inherent to the module and the parameters dependent on the connection, the robot kinematics and dynamics are modeled with a simple communication protocol. In particular, by defining the activity matrix, dynamic parameters that depend on the connected branch and connection direction are initialized to the correct values. A recursive dynamics algorithm is derived to obtain the correct torque when the joint was flexible. The software architecture is validated with the ModMan hardware for multiple arbitrary combinations of the modules.

- **Preparatory Safety Measures**

  To make the robot have a reactive posture to an object, new safety measures have been defined. From the fact that the robot is safe if it has large manipulability in the direction of interest, the distance between the first principal component of the manipulability matrix and the direction of interest is defined as a safety measure. Instead of the commonly used vector norm, the distance defined in the positive semi-definite space with rank one is used to

allow the robot to converge quickly to the position with the minimum value of the distance. By setting the direction of interest to the gradient direction of the repulsive potential function defined from the object, the robot has high manipulability in a direction reactive to the object. The distance measure has many desirable properties including invariance under scaling, well-defined near the kinematic singularity, and applicability to low-dof robots. A non-zero gradient in the vicinity of kinematic singularity guarantees a fast escape time from the singular position of the robot. Analytic formulations of all the gradients offer numerical stability and enable fast computation for real-time applications. The proposed distance measure can be applicable to many safety-related applications including collision avoidance, velocity shaping, and impedance control.

- **Collision Avoidance Algorithms with Preparatory Motion**

  Two different collision avoidance algorithms with proposed preparatory measures have been developed. The first algorithm uses a conventional potential field method to control the robot in the gradient direction the potential function with respect to joint variables. By adding preparatory motion to repulsive motion, collisions can be avoided robustly even when objects approach the robot in a dangerous direction. To avoid collisions while performing the nominal task, the algorithm smoothly blend the avoidance motion and the nominal motion using linear weights. A set of simulations and hardware experiments are performed to show that the collision avoidance with the preparatory action is superior to the repulsive-only collision avoidance.

  The second algorithm uses an invariance control framework. Also in this case, the preparatory safety measure should be included as a constraint because

when the system is constrained only with repulsive functions, the control input cannot be generated when the object approaches the robot in a dangerous direction. However, there are cases where preparatory constraints cannot generate control inputs with existing method of creating corrective control inputs. Therefore, by solving a separate optimization problem, the state of the robot can be quickly placed in an invariant set of the preparatory constraints. The algorithm is validated with a set of simulations. Simulations have shown that the algorithm can reliably avoid collisions in both safe and dangerous initial poses.

6.2 Future Work

There are a number of applications in which our contributions of this thesis can be extended. Here are some important studies.

- **Automatic Gain Tuning of Modular Manipulator**

  In the case of reconfigurable robots developed to date, the number of assembly cases is not large, so the burden of gain tuning is small. However, as the number of modules increases and the shape becomes more complex, it is difficult to perform such gain tuning in advance. Therefore, it is necessary to have an algorithm that can find appropriate gains for controllers according to the kinematic and dynamic parameters of the robot. It should be possible to adjust the gain adaptively by taking initial gain based on the dynamic parameters from the automatic dynamic modeling algorithm presented in this thesis, and properly compensating for unmodeled dynamics.

- **Safety-Guaranteed Time-Optimal Path Planning**

As described in [32], algorithms have been developed that can move faster while guaranteeing safety by controlling the robot in the direction of reducing the reflected mass. It is expected that more efficient path generation can be achieved by using the safety measures introduced in this thesis. In addition, it is expected that by adding safety constraints in time-optimal path planning framework described in [68], it will be possible to create a minimum time path that guarantees safety.

- **Tracking of Stiffness Matrix**

  The stiffness matrix at the control point of the robot also forms an ellipsoid, similar to the manipulability. In the case of microsurgical robots, the stiffness matrix is more important because it is very soft compared to industrial robots. By reducing the distance on $S^{+}(1, n)$, it is possible to control the robot in the direction of increasing or decreasing the stiffness in a specific direction.

A

Appendix

A.1 Preliminaries on Graph Theory

In what follows, we introduce basic preliminaries of graph theory outlined in [69].

Definition A.1.1. A graph $G = (V, E)$ consist of a vertex set, $V(G)$, and an edge set, $E(G)$, such that every edge in $E(g)$ is associated with a pair of vertices in $V(G)$.

Definition A.1.2. A labeled graph is a graph in which the vertices are labeled by $v_1, v_2, \cdots, v_m$ and the edges are labeled by $e_1, e_2, \cdots, e_n$, such that $V = v_1, v_2, \cdots, v_m$ and $E = e_1, e_2, \cdots, e_n$.

Definition A.1.3. Let $G = (V, E)$ be a labeled graph. The incidence matrix $X(G)$ is the $m \times n$ matrix in which the entry in row i and column j is 1, if edge e_j is incident on vertex v_i; otherwise, it is 0.

A.2 Lie-Theoretic Formulations of Robot Kinematics and Dynamics

In what follows, we introduce the Lie-theoretic formulations of kinematics and dynamics of a robot and their derivative, outlined in [46] and [70]. The task space of a robot can be defined on the space of the special Euclidean group $SE(3)$ while the 4×4 rigid body transformation matrix $T \in SE(3)$ is of the form

$$T = \begin{pmatrix} R & x \\ 0_{1\times 3} & 1 \end{pmatrix}, \tag{A.2.1}$$

where R is the 3 by 3 rotation matrix, $x \in \mathbb{R}^3$ is the position vector. For any $T \in SE(3)$, one can find corresponding screw parameter $A \in \mathbb{R}^6 = \left(\omega^T, v^T\right)^T$ with $\omega, v \in \mathbb{R}^3$, which satisfies the matrix exponential of the form

$$T = e^{[A]}. \tag{A.2.2}$$

The square bracket notation $[A]$ for screw parameter A denotes

$$[A] = \begin{pmatrix} [\omega] & v \\ 0_{1\times 3} & 0 \end{pmatrix}, \tag{A.2.3}$$

where, again, the square bracket notation $[\omega]$ for three dimensional vector $\omega = (\omega_1, \omega_2, \omega_3)^T$ is

$$[\omega] = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \tag{A.2.4}$$

The forward kinematics of a serial robot from space-fixed frame $\{s\}$ to a point c attached on the i'th link of the robot, T_{sc}, is expressed as a product of exponential form as

$$T_{sc} = \begin{pmatrix} R_{sc} & x_{sc} \\ 0 & 1 \end{pmatrix} = M_{s,1} e^{[J_1^1]} \cdots M_{i-1,i} e^{[J_i^i]} M_{i,c}, \tag{A.2.5}$$

where $J_i^i \in se(3)$ is the screw parameter of i'th joint, $M_{i-1,i} \in SE(3)$ is the initial transformation from link frame $\{i-1\}$ to $\{i\}$.

The velocity kinematics for the point c is represented as

$$V^c = J^c \dot{q}, \tag{A.2.6}$$

where $J^c \in \mathbb{R}^{6\times n}$ is the Jacobian matrix defined on the frame $\{c\}$, which is of the form

$$J^c = \begin{pmatrix} J_1^c \cdots J_i^c & 0 \cdots 0 \end{pmatrix}. \tag{A.2.7}$$

The translation part of (A.2.6) is given as

$$x_{sc} = R_{sc} J_v^c \dot{q}. \tag{A.2.8}$$

The value of $\frac{\partial R_{sc}}{\partial q_k}$ is derived from the product of exponential form of forward kinematics, given as

$$\frac{\partial R_{sc}}{\partial q_k} = [R_{sk} J_{w,k}^k] R_{sc}, \tag{A.2.9}$$

where $R_{sk} \in SO(3)$ is the orientation of link frame $\{k\}$ expressed in the $\{s\}$.

As detailed in [70], the derivative of each column of J^c is given by

$$\frac{\partial J_j^i}{\partial q_k} = \begin{cases} ad_{J_j^i} J_k^i & \text{if } j < k \\ 0 & \text{otherwise} \end{cases} \tag{A.2.10}$$

The rigid body dynamics of a serial robot is of the form

$$M(q)\ddot{q} + c(q,\dot{q}) + g(q) = \tau, \tag{A.2.11}$$

where $M(q) \in \mathbb{R}^{n\times n}$ is joint space mass matrix, $c(q,\dot{q}) \in \mathbb{R}^n$ is the Coriolis force, $g(q) \in \mathbb{R}^n$ is the gravitational force, and $\tau \in \mathbb{R}^n$ is joint torque. The joint space mass matrix can be factorized as

$$M(q) = A(q)^T G A(q), \tag{A.2.12}$$

where

$$A = \begin{pmatrix} J_1^1 & 0 & \cdots & 0 \\ J_1^2 & J_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ J_1^n & J_2^n & \cdots & J_n^n \end{pmatrix} \tag{A.2.13}$$

So the derivative of M^{-1} can be represented as

$$\frac{\partial M^{-1}}{\partial q_k} = -M^{-1}\frac{\partial M}{\partial q_k}M^{-1} \tag{A.2.14}$$

$$\frac{\partial M}{\partial q_k} = \frac{\partial A^T}{\partial q_k}GA + A^T G\frac{\partial A}{\partial q_k} \tag{A.2.15}$$

$$\frac{\partial A_{ij}}{\partial q_k} = \frac{\partial J_j^i}{\partial q_k}, \tag{A.2.16}$$

A.3 Derivatives of Eigenvectors and Eigenvalues

Throughout the thesis, computations of desired input for robots often include the derivatives of eigenvalues and their corresponding eigenvectors. In this section, we briefly review the result of [71] to see how we calculate the values for distinct eigenvalue cases.

Assume $A \in SP(n)$ has distinct n eigenvalues and can be decomposed into

$$A = V\Lambda V^T \tag{A.3.17}$$

where $\Lambda = diag(\lambda_1, \cdots, \lambda_n)$ is eigenvalue matrix and $V = (v_1, \cdots, v_n) \in SO(n)$ is eigenvector matrix where $v_i \in \mathbb{R}^n$ is eigenvector corresponding to λ_i. Following

properties are hold from definition

$$Av_i = \lambda_i v_i \tag{A.3.18}$$

$$v_i^T v_i = 1. \tag{A.3.19}$$

Assuming A is dependent on m parameters $q_1, \cdots, q_m$, the derivative of above properties with respect to k'th parameter q_k are

$$\frac{\partial A}{\partial q_k} v_i + A \frac{\partial v_i}{\partial q_k} = \frac{\partial \lambda_i}{\partial q_k} v_i + \lambda_i \frac{\partial v_i}{\partial q_k} \tag{A.3.20}$$

$$v_i^T \frac{\partial v_i}{\partial q_k} = 0 \tag{A.3.21}$$

$$\frac{\partial \lambda_i}{\partial q_k} = v_i^T \frac{\partial A}{\partial q_k} v_i \tag{A.3.22}$$

Next, we take the inner product of (A.3.20) with $v_j, i \neq j$. Symmetry of A and orthogonality of eigenvectors gives

$$v_j^T \frac{\partial v_i}{\partial q_k} = \frac{1}{\lambda_i - \lambda_j} v_j^T \frac{\partial A}{\partial q_k} v_i. \tag{A.3.23}$$

Stacking (A.3.23) and (A.3.21) for all i and left-multiply V gives

$$\frac{\partial v_i}{\partial q_k} = \sum_{j \neq i} \left(\frac{1}{\lambda_i - \lambda_j} v_j^T \frac{\partial A}{\partial q_k} v_i\right) v_j \tag{A.3.24}$$

A.4 Proof of Proposition Proposition 4.1

To see this phenomenon more in detail, we look at the problem in Cartesian space. Since $K(q)$ from (4.1.4) is 3 by 3 positive (semi-)definite, the eigen values of $K(q)$ are non-negative. Eigen value decomposition $K = Q\Lambda Q^T$ simplifies (4.1.7) as

$$\begin{aligned} \mu &= u^{cT} Q \Lambda Q^T u^c \\ &= \tilde{u}^T \Lambda \tilde{u} \end{aligned} \tag{A.4.25}$$

where $Q \in SO(3)$ is eigen vector matrix, $\tilde{u} = Q^T u^c$ is a rotated direction, $\Lambda \in \mathbb{R}^{3\times 3}$ is diagonal eigen value matrix

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}. \tag{A.4.26}$$

with $\lambda_1 > \lambda_2 > \lambda_3 \geq 0$. Equation (A.4.25) can be thought geometrically as an ellipsoid whose major axis is x-direction and minor axis is z-direction. If we represent $\tilde{u}$ in a spherical coordinate, (A.4.25) becomes

$$\mu = \cos^2\phi \left(\lambda_1 \cos^2\theta + \lambda_2 \sin^2\theta - \lambda_3\right) + \lambda_3. \tag{A.4.27}$$

where $\tilde{u} = (\cos\phi\cos\theta, \cos\phi\sin\theta, \sin\phi)^T$ with $\theta \in (0, 2\pi)$ and $\phi \in (0, 2\pi)$. Equation (A.4.27) has its local minima at $\phi = \frac{\pi}{2}$. To see the behavior on the ellipse, we find partial derivatives of μ w.r.t. ϕ and θ, respectively, from (A.4.27),

$$\frac{\partial \mu}{\partial \theta} = 2\left(\lambda_2 - \lambda_1\right)\cos^2\phi \sin\theta\cos\theta \tag{A.4.28}$$

$$\frac{\partial \mu}{\partial \phi} = -2\left(\lambda_1 \cos^2\theta + \lambda_2 \sin^2\theta - \lambda_3\right)\sin\phi\cos\phi. \tag{A.4.29}$$

Equation (A.4.28) and (A.4.29) become zero at $\phi = \frac{\pi}{2}$ due to $\cos\phi$.

One way to make a zero of function to non-zero is to use indeterminate forms, $\frac{0}{0}$. We utilize important lemma from [67]; a composition of a function with monotonically increasing function does not change the position of local minima/maxima. For a scalar function $g(x)$, the derivatives of $g \circ \mu(\theta, \phi)$ with respect to θ and ϕ are given by chain rule,

$$\frac{\partial g \circ \mu}{\partial \theta} = \frac{\partial g}{\partial \mu}\frac{\partial \mu}{\partial \theta} \tag{A.4.30}$$

$$\frac{\partial g \circ \mu}{\partial \phi} = \frac{\partial g}{\partial \mu}\frac{\partial \mu}{\partial \phi}. \tag{A.4.31}$$

If $g(x)$ is of the form of equation (4.2.21) from Proposition 1,

$$\frac{\partial g}{\partial \mu} = \alpha\gamma \left(\cos^2\phi \left(\lambda_1 \cos^2\theta + \lambda_2 \sin^2\theta - \lambda_3\right)\right)^{\gamma-1}. \tag{A.4.32}$$

Since $\left(\lambda_1 \cos^2\theta + \lambda_2 \sin^2\theta - \lambda_3\right)$ is always positive for $\lambda_1 > \lambda_2 > \lambda_3 \geq 0$, $\left(\cos^2\phi\right)^{\gamma-1}$ should approach ∞ as ϕ get closer to $\frac{\pi}{2}$ and $g(x)$ should be monotonically increasing function, which gives condition $0 < \gamma < 1$. Then the limits of equation (A.4.30) and (A.4.31) are simplified as

$$\lim_{\phi \to \frac{\pi}{2} \pm 0} \frac{\partial g(\mu)}{\partial \mu}\frac{\partial \mu}{\partial \theta} = C_\theta \lim_{\phi \to \frac{\pi}{2} \pm 0} \cos\phi^{2\gamma} \tag{A.4.33}$$

$$\lim_{\phi \to \frac{\pi}{2} \pm 0} \frac{\partial g(\mu)}{\partial \mu}\frac{\partial \mu}{\partial \phi} = C_\phi \lim_{\phi \to \frac{\pi}{2} \pm 0} \cos\phi^{2\gamma-1}, \tag{A.4.34}$$

where C_θ and C_ϕ are finite constants. Equation A.4.33 is always zero for $0 < \gamma < 1$, while A.4.34 is not zero when $0 < \gamma \leq 0.5$.

Now we turn our attention to joint variable q. By chain rule,

$$\frac{\partial g(\mu)}{\partial q_k} = \frac{\partial g(\mu)}{\partial \mu}\left(\frac{\partial \mu}{\partial \phi}\frac{\partial \phi}{\partial q_k} + \frac{\partial \mu}{\partial \theta}\frac{\partial \theta}{\partial q_k} + \sum_{i=1}^{3}\frac{\partial \mu}{\partial \lambda_i}\frac{\partial \lambda_i}{\partial q_k}\right). \tag{A.4.35}$$

where

$$\lim_{\phi \to \frac{\pi}{2} \pm 0} \frac{\partial g(\mu)}{\partial \mu}\frac{\partial \mu}{\partial \lambda_i} = C_{\lambda_i} \lim_{\phi \to \frac{\pi}{2} \pm 0} \cos\phi^{2\gamma}, \tag{A.4.36}$$

which are zeros for $i = 1, 2, 3$. Then A.4.35 is simply $\frac{\partial V_p}{\partial q_k} = \frac{\partial V_p}{\partial \phi}\frac{\partial \phi}{\partial q_k}$.

In terms of manipulability, $\frac{\partial \phi}{\partial q_k}$ at $\phi = \frac{\pi}{2}$ represents the ability to move orthogonal to minor axes and it is determined by θ. If $\lambda_1 > 0$, there exist θ that generate linear velocity in the direction orthogonal to its minor axis, so A.4.35 is not zero.

When $\gamma = 0.5$, $g(x) = \sqrt{x - \lambda_3}$ which results in finite value of A.4.34. □

A.5 Proof of Triangle Inequality When $p = 1$

Given three matrices $A_1, A_2, A_3 \in S^+(1, n)$ in form of $A_i = \lambda_i u_i u_i^T$, the length (4.3.38) satisfies the triangle inequality

$$l_{1,2} + l_{2,3} \geq l_{3,1} \tag{A.5.37}$$

where

$$l_{i,j} = \sqrt{\theta_{i,j}^2 + k\left(\log \frac{\lambda_i}{\lambda_j}\right)^2} \tag{A.5.38}$$

and $\theta_{i,j} = \cos^{-1}(u_i^T u_j)$.

Proof. For simplicity, we denote that $\sqrt{k}\log\frac{\lambda_1}{\lambda_2} = X$ and $\sqrt{k}\log\frac{\lambda_2}{\lambda_3} = Y$. Then above equation become

$$\sqrt{\theta_{1,2}^2 + X^2} + \sqrt{\theta_{2,3}^2 + Y^2} \geq \sqrt{\theta_{3,1}^2 + (X+Y)^2}. \tag{A.5.39}$$

Since the angle part and the logarithm part are the distance on each space, i.e., $Gr(1, n)$ and $P(1)$, following triangle inequality hold.

$$\theta_{1,2} + \theta_{2,3} \geq \theta_{3,1} \tag{A.5.40}$$

$$2\theta_{1,2}\theta_{2,3} \geq \theta_{3,1}^2 - \theta_{1,2}^2 - \theta_{2,3}^2 \tag{A.5.41}$$

$$|X| + |Y| \geq |X + Y|. \tag{A.5.42}$$

Taking square on both side of (A.5.39) and rearranging the equation gives

$$2\sqrt{\left(\theta_{1,2}^2 + X^2\right)\left(\theta_{2,3}^2 + Y^2\right)} \geq 2XY + \theta_{3,1}^2 - \theta_{1,2}^2 - \theta_{2,3}^2 \tag{A.5.43}$$

From (A.5.41) it becomes

$$2\sqrt{\left(\theta_{1,2}^2 + X^2\right)\left(\theta_{2,3}^2 + Y^2\right)} \geq 2XY + 2\theta_{1,2}\theta_{2,3}. \tag{A.5.44}$$

Taking another square on both side gives

$$(\theta_{1,2}Y - \theta_{2,3}X) \geq 0 \tag{A.5.45}$$

□

A.6 Detailed Conditions for a Danger Field

We present the detailed conditions for a scalar function to be a danger field, as described in [44]. Let $x_{\text{rel}} = x_{\text{obj1}} - x_{\text{obj2}}$ and $\dot{x}_{\text{rel}} = \dot{x}_{\text{obj1}} - \dot{x}_{\text{obj2}}$ be a relative position and velocity between two objects, respectively, and $\varphi = \angle(x_{\text{rel}}, \dot{x}_{\text{rel}}) \in [-\pi, \pi)$ be the angle between x_{rel} and $\dot{x}_{\text{rel}}$.

Definition A.6.1. A differentiable scalar function $V_r(x_{\text{rel}})$ is called a static danger field if it satisfies the conditions:

- $\exists f_r : \mathbb{R}^+ \to \mathbb{R}^+$, such that $V_r(x_{\text{rel}}) \equiv f_r(\|x_{\text{rel}}\|)$.
- $\frac{df_r(\|x_{\text{rel}}\|)}{d\|x_{\text{rel}}\|} < 0, \forall \|x_{\text{rel}}\| > 0$

  It is simply a radial potential function defined in [60].

Definition A.6.2. A differentiable scalar function $V_r(x_{\text{rel}}, \dot{x}_{\text{rel}})$ is called a kineto-static danger field if it satisfies the conditions:

- $\exists f_r : \mathbb{R}^3 \to \mathbb{R}^+$, such that $V_r(x_{\text{rel}}, \dot{x}_{\text{rel}}) \equiv f_r(\|x_{\text{rel}}\|, \|\dot{x}_{\text{rel}}\|, \varphi)$.
- $V_r(x_{\text{rel}}, 0)$ is a static danger field.
- $\frac{df_r(\|x_{\text{rel}}\|, \|\dot{x}_{\text{rel}}\|, \varphi)}{d\|x_{\text{rel}}\|} < 0, \forall \|x_{\text{rel}}\| > 0, \forall \|\dot{x}_{\text{rel}}\| \geq 0, \forall \varphi \in [-\pi, \pi)$.
- $\frac{df_r(\|x_{\text{rel}}\|, \|\dot{x}_{\text{rel}}\|, \varphi)}{d\|\dot{x}_{\text{rel}}\|} > 0, \forall \|x_{\text{rel}}\| > 0, \forall \|\dot{x}_{\text{rel}}\| \geq 0, \forall \varphi \in (-\frac{\pi}{2}, \frac{\pi}{2})$.

- $\varphi \frac{df_r(\|x_{\text{rel}}\|, \|\dot{x}_{\text{rel}}\|, \varphi)}{d\varphi} < 0, \forall \|x_{\text{rel}}\| > 0, \forall \|\dot{x}_{\text{rel}}\| > 0, \forall \varphi \in [-\pi, \pi)$.

That is, the value of the danger field becomes larger as the size of the relative position decreases, the size of the relative speed increases, and the angle between the two vectors decreases.

Bibliography

[1] Toshio Fukuda and Seiya Nakagawa. Dynamically reconfigurable robotic system. In *1988 IEEE International Conference on Robotics and Automation*, pages 1581–1586. IEEE, 1988.

[2] Donald Schmitz, Pradeep Khosla, and Takeo Kanade. The CMU reconfigurable modular manipulator system. 1988.

[3] Jinguo Liu, Xin Zhang, and Guangbo Hao. Survey on research and development of reconfigurable modular robots. *Advances in Mechanical Engineering*, 8(8):1–21, 2016.

[4] Chris Paredis and Pradeep Khosla. RMMS: Reconfigurable modular manipulator system project. In *Video Proceedings of the 1997 IEEE International Conference on Robotics and Automation*. IEEE, January 1997.

[5] Christiaan J.J. Paredis, H. Benjamin Brown, and Pradeep K. Khosla. A rapidly deployable manipulator system. *Robotics and Autonomous Systems*, 21(3):289–304, 1997.

[6] Schunk Gmbh & Co. KG. Dextrous lightweight arm LWA 4D. [Online] Available: https://schunk.com/fileadmin/pim/docs/im0012315.pdf, 2010.

[7] Robert R. Burridge, Mars W. Chu, and Bryn T. Wolfe. Apparatus, systems, and methods for reconfigurable robotic manipulator and coupling, December 2013. US Patent App. 13/911,073.

[8] Nestor Gonzalez Lopez, Yue Leire Erro Nuin, Elias Barba Moral, Lander Usategui San Juan, Alejandro Solano Rueda, Víctor Mayoral Vilches, and Risto

Kojcev. gym-gazebo2, a toolkit for reinforcement learning using ROS 2 and Gazebo. *arXiv preprint arXiv:1903.06278*, 2019.

[9] Wael Saab, Peter Racioppo, and Pinhas Ben-Tzvi. A review of coupling mechanism designs for modular reconfigurable robots. *Robotica*, 37(2):378–403, 2019.

[10] Mark Yim, Ying Zhang, Kimon Roufas, David Duff, and Craig Eldershaw. Connecting and disconnecting for chain self-reconfiguration with PolyBot. *IEEE/ASME Transactions on Mechatronics*, 7(4):442–451, 2002.

[11] John W. Suh, Samuel B. Homans, and Mark Yim. Telecubes: Mechanical design of a module for self-reconfigurable robotics. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 4, pages 4095–4101. IEEE, 2002.

[12] Andreas Lyder, Ricardo Franco Mendoza Garcia, and Kasper Stoy. Genderless connection mechanism for modular robots introducing torque transmission between modules. In *Proceedings of the ICRA Workshop on Modular Robots, State of the Art*, pages 77–81, 2010.

[13] Guoqiang Fu, Arianna Menciassi, and Paolo Dario. Development of a genderless and fail-safe connection system for autonomous modular robots. In *2011 IEEE International Conference on Robotics and Biomimetics*, pages 877–882. IEEE, 2011.

[14] S. G. M. Hossain, Carl A. Nelson, and Prithviraj Dasgupta. RoGenSiD: A rotary plate genderless single-sided docking mechanism for modular self-reconfigurable robots. In *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*.

American Society of Mechanical Engineers Digital Collection, 2013.

[15] Christopher Parrott, Tony J. Dodd, and Roderich Groß. HiGen: A high-speed genderless mechanical connection mechanism with single-sided disconnect for self-reconfigurable modular robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3926–3932. IEEE, 2014.

[16] Christopher Parrott, Tony J. Dodd, and Roderich Groß. HyMod: A 3-DOF hybrid mobile and self-reconfigurable modular robot and its extensions. In *Distributed Autonomous Robotic Systems*, pages 401–414. Springer, 2018.

[17] Sami Haddadin, Alessandro De Luca, and Alin Albu-Schäffer. Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics*, 33(6):1292–1312, 2017.

[18] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous Robot Vehicles*, pages 396–404. Springer, 1986.

[19] Oliver Brock, Oussama Khatib, and Sriram Viji. Task-consistent obstacle avoidance and motion behavior for mobile manipulation. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 1, pages 388–393. IEEE, 2002.

[20] Alexander Dietrich, Thomas Wimbock, Alin Albu-Schäffer, and Gerd Hirzinger. Integration of reactive, torque-based self-collision avoidance into a task hierarchy. *IEEE Transactions on Robotics*, 28(6):1278–1293, 2012.

[21] Fabrizio Flacco, Torsten Kroeger, Alessandro De Luca, and Oussama Khatib. A depth space approach for evaluating distance to objects. *Journal of Intelligent & Robotic Systems*, 80(1):7–22, 2015.

[22] Matteo P. Polverini, Andrea M. Zanchettin, and Paolo Rocco. A computationally efficient safety assessment for collaborative robotics applications. *Robotics and Computer-Integrated Manufacturing*, 46:25–37, 2017.

[23] Melanie Kimmel and Sandra Hirche. Invariance control for safe human-robot interaction in dynamic environments. *IEEE Transactions on Robotics*, 33(6):1327–1342, 2017.

[24] Oussama Khatib. Inertial properties in robotic manipulation: An object-level framework. *The International Journal of Robotics Research*, 14(1):19–36, 1995.

[25] Stephen L. Chiu. Task compatibility of manipulator postures. *The International Journal of Robotics Research*, 7(5):13–21, 1988.

[26] Khiar Nait-Chabane, Sébastien Delarue, Philippe Hoppenot, and Etienne Colle. Strategy of approach for seizure of an assistive mobile manipulator. *Robotics and Autonomous Systems*, 57(2):222–235, 2009.

[27] Ralf Koeppe and Tsueno Yoshikawa. Dynamic manipulability analysis of compliant motion. In *Proceedings 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovative Robotics for Real-World Applications*, volume 3, pages 1472–1478. IEEE, 1997.

[28] Ian D. Walker. Impact configurations and measures for kinematically redundant and multiple armed robot systems. *IEEE Transactions on Robotics and Automation*, 10(5):670–683, 1994.

[29] Sungchul Kang, Kiyoshi Komoriya, Kazuhito Yokoi, Tetsuo Koutoku, and Kazuo Tanie. Reduced inertial effect in damping-based posture control of mobile manipulator. In *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 488–493. IEEE, 2001.

[30] Rossi Rossi, Matteo P. Polverini, Andrea M. Zanchettin, and Paolo Rocco. A pre-collision control strategy for human-robot interaction based on dissipated energy in potential inelastic impacts. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 26–31. IEEE, 2015.

[31] Jochen Heinzmann and Alex Zelinsky. Quantitative safety guarantees for physical human-robot interaction. *The International Journal of Robotics Research*, 22(7-8):479–504, 2003.

[32] Sami Haddadin, Simon Haddadin, Augusto Khoury, Tim Rokahr, Sven Parusel, Rainer Burgkart, Antonio Bicchi, and Alin Albu-Schäffer. On making robots understand safety: Embedding injury knowledge into control. *The International Journal of Robotics Research*, 31(13):1578–1602, 2012.

[33] Nico Mansfeld, Badis Djellab, Jaime R. Veuthey, Fabian Beck, Christian Ott, and Sami Haddadin. Improving the performance of biomechanically safe velocity control for redundant robots through reflected mass minimization. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5390–5397. IEEE, 2017.

[34] Nicholas A. Scott and Carignan R. Carignan. A line-based obstacle avoidance technique for dexterous manipulator operations. In *2008 IEEE International Conference on Robotics and Automation*, pages 3353–3358. IEEE, 2008.

[35] Jorgen Sverdrup-Thygeson, Signe Moe, Kristin Ytterstad Pettersen, and Jan Tommy Gravdahl. Kinematic singularity avoidance for robot manipulators using set-based manipulability tasks. In *2017 IEEE Conference on Control Technology and Applications*, pages 142–149. IEEE, 2017.

[36] Paolo Di Lillo, Stefano Chiaverini, and Gianluca Antonelli. Handling robot constraints within a set-based multi-task priority inverse kinematics framework. In *2019 IEEE International Conference on Robotics and Automation*, pages 7477–7483. IEEE, 2019.

[37] Noémie Jaquier, Leonel Dario Rozo, Darwin G. Caldwell, and Sylvain Calinon. Geometry-aware tracking of manipulability ellipsoids. In *Robotics: Science and Systems*, 2018.

[38] Laura Kelmar and Pradeep K. Khosla. Automatic generation of kinematics for a reconfigurable modular manipulator system. In *1988 IEEE International Conference on Robotics and Automation*, pages 663–668. IEEE, 1988.

[39] I-Ming Chen and Guilin Yang. Automatic generation of dynamics for modular robots with hybrid geometry. In *1997 IEEE International Conference on Robotics and Automation*, volume 3, pages 2288–2293. IEEE, 1997.

[40] Andrea Giusti and Matthias Althoff. On-the-fly control design of modular robot manipulators. *IEEE Transactions on Control Systems Technology*, 26(4):1484–1491, 2017.

[41] Zhuming Bi, Yingzi Lin, and Wenjun Zhang. The general architecture of adaptive robotic systems for manufacturing applications. *Robotics and Computer-Integrated Manufacturing*, 26(5):461–470, 2010.

[42] Silvere Bonnabel and Rodolphe Sepulchre. Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1055–1070, 2010.

[43] Jan Wolff and Martin Buss. Invariance control design for constrained nonlinear systems. *IFAC Proceedings Volumes*, 38(1):37–42, 2005.

[44] Bakir Lacevic and Paolo Rocco. Kinetostatic danger field-a novel safety assessment for human-robot interaction. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2169–2174. IEEE, 2010.

[45] Matteo P. Polverini, Andrea M. Zanchettin, and Paolo Rocco. Real-time collision avoidance in human-robot interaction based on kinetostatic safety field. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4136–4141. IEEE, 2014.

[46] Kevin M. Lynch and Frank C. Park. *Modern robotics: Mechanics, planning, and control*. Cambridge University Press, 2017.

[47] Woosub Lee, Seonghun Hong, Sungchul Kang, and Kanggyun Kim. Module connection mechanism capable of genderless coupling, December 2017. US Patent 9853386B1.

[48] Universal Robot. UR5 technical specifications. [Online] Available: https://www.universal-robots.com/media/50588/ur5_en.pdf, 2016.

[49] Tsuneo Yoshikawa. Manipulability of robotic mechanisms. *The International Journal of Robotics Research*, 4(2):3–9, 1985.

[50] J. Kenneth Salisbury and John J. Craig. Articulated hands: Force control and kinematic issues. *The International Journal of Robotics Research*, 1(1):4–17, 1982.

[51] I-Ming Chen and Guilin Yang. Automatic model generation for modular reconfigurable robot dynamics. 1998.

[52] Frank C. Park, James E. Bobrow, and Scott R. Ploen. "a Lie group formulation of robot dynamics". *The International Journal of Robotics Research*, 14(6):609–618, 1995.

[53] Alessandro De Luca and Wayne J. Book. Robots with flexible elements. In *Springer Handbook of Robotics*, pages 243–282. Springer, 2016.

[54] Joo-Haeng Lee, Woo-Han Yun, Jaeyeon Lee, and Jaehong Kim. Synthetic learning set for object pose estimation: Initial experiments. In *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence*, pages 106–108. IEEE, 2017.

[55] Apple. ARKit. [Online] Available: https://developer.apple.com/augmented-reality/arkit/.

[56] Brian V. Mirtich. *Impulse-based dynamic simulation of rigid body systems.* PhD thesis, University of California at Berkeley, 1996.

[57] Morteza Azad, Jan Babič, and Michael Mistry. Effects of the weighting matrix on dynamic manipulability of robots. *Autonomous Robots*, 43(7):1867–1879, 2019.

[58] Alan Edelman, Tomás A Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[59] Maher Moakher and Philipp G. Batchelor. Symmetric positive-definite matrices: From geometry to applications and visualization. In *Visualization and Processing of Tensor Fields*, pages 285–298. Springer, 2006.

[60] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[61] Jonghoon Park, Wangkyun Chung, and Youngil Youm. Computation of gradient of manipulability for kinematically redundant manipulators including dual manipulators system. *Transactions on Control, Automation and Systems Engineering*, 1(1):8–15, 1999.

[62] Neuromeka company. Indy7. [Online] Available: https://www.neuromeka.com /cobot.

[63] Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[64] Hassan K. Khalil and Jessy W. Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.

[65] Melanie Kimmel and Sandra Hirche. Active safety control for dynamic human-robot interaction. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4685–4691. IEEE, 2015.

[66] Michael Scheint, Jan Wolff, and Martin Buss. Invariance control in robotic applications: Trajectory supervision and haptic rendering. In *2008 American Control Conference*, pages 1436–1442. IEEE, 2008.

[67] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[68] Quang-Cuong Pham. A general, fast, and robust implementation of the time-optimal path parameterization algorithm. *IEEE Transactions on Robotics*, 30(6):1533–1540, 2014.

[69] John A. Bondy and Uppaluri S. R. Murty. *Graph theory with applications*, volume 290. Macmillan London, 1976.

[70] Sung-Hee. Lee, Junggon Kim, Frank C. Park, Munsang Kim, and James E. Bobrow. Newton-type algorithms for dynamics-based robot movement optimization. *IEEE Transactions on Robotics*, 21(4):657–667, 2005.

[71] Jan R Magnus. On differentiating eigenvalues and eigenvectors. *Econometric Theory*, 1(2):179–191, 1985.

국문초록

다음 세대의 로봇은 사람과 가까이에서 협업할 수 있는 기능을 가져야한다. 그와 동시에, 로봇은 다양하게 변하는 작업에 대해 유연하게 대처할 수 있도록 자신의 구조를 바꾸는 기능을 가져야 한다. 이러한 두 가지 요구조건을 만족시키기 위해, 본 논문에서는 새로운 모듈라 로봇 시스템과 프로그래밍 아키텍쳐를 제시하고, 사람이 존재하는 동적 환경에서 안전한 로봇의 운용을 위한 실시한 경로 계획 알고리즘을 제시한다.

개발된 모듈라 로봇의 두 가지 핵심적인 혁신성은 무성별 커넥터와 다자유도 모듈에서 찾을 수 있다. 입력/출력 방향에 상관 없이 모듈이 연결될 수 있도록 함으로써, 무성별 커넥터는 결합 가능한 경우의 수를 늘릴 수 있다. 개발된 무성별 커넥터는 산업용 로봇에서 요구되는 충분한 부하를 견딜 수 있도록 설계되었다. 2 자유도 모듈의 설계에서 두 축 사이에 오프셋을 가지도록 함으로써 전체적인 완성도 및 안전도를 증가시켰다.

무성별 커넥터와 다자유도 모듈로 인한 모델링의 복잡성에 대응하기 위해, 일반적인 모듈라 로봇을 위한 소프트웨어 아키텍쳐를 제안하였다. 기존 모듈라 로봇의 연결을 나타내는 방법이 모든 링크와 조인트 사이의 연결 관계를 별도로 나타내야하는 것과 다르게, 제안된 아키텍쳐는 모듈들 사이의 연결관계만을 나타냄으로써 효율적인 다자유도 모듈의 연결관계를 나타낼 수 있다는 것을 특징으로 한다. 이를 위해 트리 구조를 가지는 일반적인 다자유도 모듈의 성질을 나타내는 데이터 구조를 정의하였다. 모듈들 사이의 연결관계 및 데이터 구조를 이용하여, 정확한 기구학/동역학 모델 파라미터를 얻어내는 순방향 재귀 알고리즘을 구현하였다.

모듈라 로봇의 안전한 운용을 위해, 기구학적 특이점에 강건한 실시간 충돌회피 알고리즘을 제안하였다. 방향성 안전도를 줄이는 방향의 제어 입력을 생성하여 물체 방향으로의 로봇 방향성 매니퓰러빌리티를 증가시키는 것이 제안한 알고리즘의 핵심이다. 기존의 방향성 안전도가 기구학적 특이점 근처에서 원하지 않는 성질을 가지는

것과는 반대로, 제안한 기하학적 안전도는 전체 조인트 공간에서 안정적인 제어 입력을 생성한다. 이 기하학적 안전도를 이용하여, 기구학적 특이점에 강건한 계층적 충돌회피 알고리즘을 구현하였다.

수학적으로 로봇의 안전도를 보장하기 위해, 상대속도에 종속적인 안전 제약조건을 가지는 불변 제어 프레임워크을 이용하여 또 하나의 충돌 회피 알고리즘을 제안하였다. 물체가 특이점 방향으로부터 로봇에 접근할 때, 로봇의 초기 상태에서 안전 제약조건을 만족시키지 못하게 되어 불변제어를 적용할 수 없게 된다. 준비 제약조건을 빠르게 임계점 아래로 감소시키는 알고리즘을 적용함으로써, 로봇은 제약조건 집합에 다시 들어가고 불변 제어 방법을 이용하여 충돌을 회피할 수 있게 된다.

개발된 재구성 로봇의 모듈라리티와 안전도는 일련의 시뮬레이션과 하드웨어 실험을 통해 검증되었다. 실시간으로 조립된 로봇의 기구학/동역학 모델을 얻어내 정밀 제어에 사용하였다. 안전한 모듈 디자인과 충돌 회피 등의 고차원 안전 기능을 통하여, 개발된 재구성 로봇은 기존 협동로봇보다 넓은 안전한 작업공간과 작업속도를 가진다.

주요어: 재구성 로봇, 모듈라 로봇 소프트웨어, 방향성 안전도, 반영 질량, 충돌 회피, 특이점 회피.

학번: 2010-23223